9.2

IBM MQ schützen

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen” auf Seite 737 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9 Release 2 von IBM® MQ und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Wenn Sie Informationen an IBMsenden, erteilen Sie IBM ein nicht ausschließliches Recht, die Informationen in beliebiger Weise zu verwenden oder zu verteilen, ohne dass eine Verpflichtung für Sie entsteht.

Inhaltsverzeichnis

SicherungIBM MQ

Sicherheit ist ein wichtiges Anliegen für Entwickler von IBM MQ-Anwendungen sowie für IBM MQ-Systemadministratoren.

Sicherheitsupdates

Stellen Sie sicher, dass alle Hardware und Software innerhalb der sicheren Zone und auf Bedienerarbeitsplätzen innerhalb ihres Support-Lebenszyklus ausgeführt wurden, mit obligatorischen Software-Updates aktualisiert wurden und Sicherheitsupdates sofort angewendet haben.

Lesen Sie weitere Informationen zu Sicherheitsupdates für folgende Produkte:

- Alle Plattformen unter IBM Security Bulletins
- APARs für Sicherheit und Systemintegrität unter z/OS auf dem IBM Z System Integrity-Portal.

Sicherheit - Übersicht

In dieser Themensammlung werden die IBM MQ-Sicherheitskonzepte vorgestellt.

Sicherheitskonzepte und -mechanismen, wie sie für alle Computersysteme gelten, werden zuerst dargestellt, gefolgt von einer Beschreibung dieser Sicherheitsmechanismen, wenn sie in IBM MQ implementiert sind.

Sicherheitskonzepte und -mechanismen

Diese Themensammlung beschreibt Sicherheitsaspekte, die Ihre IBM MQ-Installation betreffen.

Die allgemein akzeptierten Sicherheitsaspekte sind wie folgt:

- „Identifikation und Authentifizierung” auf Seite 8
- „Berechtigung” auf Seite 8
- „Prüfprotokollierungs” auf Seite 9
- „Vertraulichkeit” auf Seite 9
- „Datenintegrität” auf Seite 9

Sicherheitsmechanismen sind technische Tools und Techniken, die für die Implementierung von Sicherheitsservices verwendet werden. Ein Mechanismus kann von sich selbst oder mit anderen betrieben werden, um einen bestimmten Service bereitzustellen. Beispiele für allgemeine Sicherheitsmechanismen sind:

- „Kryptografie” auf Seite 10
- „Nachrichtendigests und digitale Signaturen” auf Seite 11
- „Digitale Zertifikate” auf Seite 12
- „Public Key Infrastructure (PKI)” auf Seite 17

Wenn Sie eine IBM MQ-Implementierung planen, können Sie angeben, welche Sicherheitsmechanismen Sie benötigen, um die Sicherheitsaspekte zu implementieren, die für Sie von Bedeutung sind. Informationen darüber, was Sie nach dem Lesen dieser Themen in Betracht ziehen sollten, finden Sie in „Sicherheitsanforderungen planen” auf Seite 90.

Zugehörige Konzepte

„Mit SSL/TLS arbeiten” auf Seite 294
In diesen Abschnitten finden Sie Anweisungen zum Ausführen von einzelnen Tasks im Zusammenhang mit der Verwendung von TLS mit IBM MQ.

Zugehörige Tasks

Verbinden von zwei WS-Managern mit TLS

Identifikation und Authentifizierung

Identifikation ist die Fähigkeit, eindeutig einen Benutzer eines Systems oder einer Anwendung zu identifizieren, die im System ausgeführt wird. *Authentifizierung* ist die Möglichkeit, zu beweisen, dass ein Benutzer oder eine Anwendung wirklich die Person oder die Anwendung ist, die/der die Anwendung beansprucht.

Beispiel: Ein Benutzer, der sich bei einem System anmeldet, indem er eine Benutzer-ID und ein Kennwort eingibt. Das System verwendet die Benutzer-ID, um den Benutzer zu identifizieren. Das System authentifiziert den Benutzer zum Zeitpunkt der Anmeldung, indem es überprüft, ob das angegebene Kennwort korrekt ist.

Fälschungssicherer Herkunftsnachweis

Der Service für den *fälschungssicheren Herkunftsnachweis* kann als Erweiterung für den Identifizierungs- und Authentifizierungsservice angezeigt werden. Im Allgemeinen gilt der fälschungssichere Herkunftsnachweis, wenn Daten elektronisch übermittelt werden, z. B. eine Bestellung an einen Börsenmakler, um Aktien zu kaufen oder zu verkaufen, oder eine Bestellung an eine Bank, um Geldbeträge von einem Konto auf ein anderes zu transferieren.

Das übergeordnete Ziel des Service für den fälschungssicheren Herkunftsnachweis ist es, zu beweisen, dass eine bestimmte Nachricht einer bestimmten Person zugeordnet ist.

Der Service für den fälschungssicheren Herkunftsnachweis kann mehr als eine Komponente enthalten, wobei jede Komponente eine andere Funktion bereitstellt. Wenn der Absender einer Nachricht das Senden einer Nachricht abweist, kann der Service für den fälschungssicheren Herkunftsnachweis mit *Ursprungsnachweis* dem Empfänger unbestreitbare Beweise liefern, dass die Nachricht von dieser bestimmten Person gesendet wurde. Wenn der Empfänger einer Nachricht jemals den Empfang dieser Nachricht verweigert, kann der Service für den fälschungssicheren Herkunftsnachweis mit *Zustellnachweis* dem Absender unleugbare Beweise liefern, dass die Nachricht von dieser bestimmten Person empfangen wurde.

In der Praxis ist ein Beweis mit nahezu 100%iger Gewissheit oder unbestreitbarer Beweislage ein schwieriges Ziel. In der realen Welt ist nichts völlig sicher. Die Verwaltung der Sicherheit ist eher mit der Verwaltung von Risiken für ein für das Geschäft akzeptables Maß verbunden. In einem solchen Umfeld ist eine realistischere Erwartung des Service für den fälschungssicheren Herkunftsnachweis in der Lage, Beweismittel bereitzustellen, die zulässig sind, und unterstützt Ihren Fall in einem Gericht.

Bei dem fälschungssicherer Herkunftsnachweis handelt es sich in einer IBM MQ-Umgebung um einen relevanten Sicherheitsservice, da IBM MQ für die elektronische Datenübertragung eingesetzt wird. Sie können z. B. zeitgleiche Angaben machen, dass eine bestimmte Nachricht von einer Anwendung gesendet oder empfangen wurde, die einer bestimmten Person zugeordnet ist.

IBM MQ mit Advanced Message Security stellt den Service für den fälschungssicheren Herkunftsnachweis nicht als Teil seiner Basisfunktionen bereit. Diese Produktdokumentation enthält allerdings einige Vorschläge dazu, wie Sie Ihren eigenen Service für den fälschungssicheren Herkunftsnachweis in einer IBM MQ-Umgebung bereitstellen können, indem Sie Ihre eigenen Exitprogramme schreiben.

Zugehörige Konzepte

„Identifikation und Authentifizierung in IBM MQ” auf Seite 24
In IBM MQ können Sie die Identifikation und Authentifizierung mithilfe von Informationen zum Nachrichtenkontext und einer gegenseitigen Authentifizierung implementieren.

Berechtigung

Berechtigung schützt kritische Ressourcen in einem System, indem der Zugriff nur auf berechtigte Benutzer und deren Anwendungen beschränkt wird. Sie verhindert die unbefugte Verwendung einer Ressource oder die Verwendung einer Ressource in einer nicht autorisierten Weise.

Zugehörige Konzepte

„Berechtigung in IBM MQ" auf Seite 24
Sie können Berechtigungen verwenden, um die Möglichkeiten von einzelnen Benutzern oder Anwendungen in Ihrer IBM MQ-Umgebung zu begrenzen.

Prüfprotokollierungs

Prüfung ist der Prozess der Aufzeichnung und Überprüfung von Ereignissen, um festzustellen, ob eine unerwartete oder unberechtigte Aktivität stattgefunden hat oder ob versucht wurde, eine solche Aktivität durchzuführen.

Weitere Informationen zum Einrichten von Berechtigungen finden Sie im Abschnitt „Planungsberechtigung" auf Seite 93 und den zugehörigen Unterabschnitten.

Zugehörige Konzepte

„Prüfung in IBM MQ" auf Seite 25
IBM MQ kann Ereignisnachrichten ausgeben, um zu erfassen, dass eine ungewöhnliche Aktivität stattgefunden hat.

Vertraulichkeit

Der Service *Vertraulichkeit* schützt sensible Informationen vor unbefugter Offenlegung.

Wenn sensible Daten lokal gespeichert werden, können die Zugriffssteuerungsmechanismen ausreichen, um sie unter der Voraussetzung zu schützen, dass die Daten nicht gelesen werden können, wenn auf sie nicht zugegriffen werden kann. Wenn ein höheres Maß an Sicherheit erforderlich ist, können die Daten verschlüsselt werden.

Verschlüsseln Sie sensible Daten, wenn sie über ein Kommunikationsnetz übertragen werden, insbesondere über ein unsicheres Netzwerk wie das Internet. In einer Netzumgebung sind die Zugriffssteuerungsmechanismen nicht wirksam gegen Versuche, die Daten abzufangen, wie z. B. die Verwittung.

Datenintegrität

Der *Datenintegritätsdienst* stellt fest, ob unbefugte Änderungen an Daten vorgenommen wurden.

Es gibt zwei Möglichkeiten, wie es zu Datenänderungen kommen kann: Einmal versehentliche Änderungen, die durch Hardware- oder Übertragungsfehler entstanden sind, oder Änderungen aufgrund eines gezielten Hackerangriffs. Viele Hardwareprodukte und Übertragungsprotokolle verfügen über Mechanismen, mit denen Hardware- und Übertragungsfehler erkannt und behoben werden können. Daher soll der Datenintegritätsdienst gezielte Angriffe erkennen.

Der Datenintegritätsdienst soll nur feststellen, ob Daten geändert wurden. Er stellt jedoch nicht den Originalzustand geänderter Daten wieder her.

Die Zugriffssteuerung kann den Datenintegritätsdienst ergänzen, da Daten, die vor Zugriffen geschützt sind, nicht geändert werden können. Wie der Vertraulichkeitsdienst bietet jedoch auch die Zugriffssteuerung keinen effizienten Schutz in einer Netzumgebung.

Verschlüsselungskonzepte

In dieser Themensammlung werden die Konzepte der Verschlüsselung beschrieben, die für IBM MQ gültig sind.

Der Begriff *Entität* bezieht sich auf einen Warteschlangenmanager, einen IBM MQ MQI client, einen einzelnen Benutzer oder auf jedes andere System, mit dem Nachrichten ausgetauscht werden können.

Zugehörige Konzepte

„Verschlüsselung in IBM MQ" auf Seite 26
In IBM MQ wird die Verschlüsselung mit dem TLS-Protokoll (Transport Security Layer) bereitgestellt.

Kryptografie

Bei der Verschlüsselung handelt es sich um den Konvertierungsprozess zwischen lesbarem Text, dem so genannten *Klartext* , und einem nicht lesbaren Format mit dem Namen *chiffriktext* .

Dies geschieht wie folgt:

1. Der Absender konvertiert die unverschlüsselte Nachricht in den Chiffriertext. Dieser Teil des Prozesses wird als *Verschlüsselung* (manchmal auch *Verschlüsselung*) bezeichnet.
2. Der Chiffriertext wird an den Empfänger übertragen.
3. Der Empfänger konvertiert die verschlüsselte Textnachricht zurück in das unverschlüsselte Textformular. Dieser Teil des Prozesses wird als *Entschlüsselung* (manchmal *Dezipherment*) bezeichnet.

Die Konvertierung umfasst eine Folge von mathematischen Operationen, die die Darstellung der Nachricht während der Übertragung ändern, sich jedoch nicht auf den Inhalt auswirken. Kryptographische Verfahren gewährleisten die Vertraulichkeit und den Schutz von Nachrichten vor unberechtigter Anzeige (Abhören), da eine verschlüsselte Nachricht nicht verständlich ist. Digitale Signaturen, die eine Zusicherung der Nachrichtenintegrität bieten, verwenden Verschlüsselungsverfahren. Weitere Informationen finden Sie unter „Digitale Signaturen in SSL/TLS” auf Seite 22.

Kryptografische Verfahren beinhalten einen allgemeinen Algorithmus, der durch die Verwendung von Schlüsseln spezifisch gemacht wird. Es gibt zwei Klassen von Algorithmen:

- Jene, die beide Parteien benötigen, um denselben geheimen Schlüssel zu verwenden. Algorithmen, die einen gemeinsam genutzten Schlüssel verwenden, werden als *symmetrische* Algorithmen bezeichnet. Abbildung 1 auf Seite 10 zeigt die symmetrische Verschlüsselung von Schlüsseln an.
- Diejenigen, die einen Schlüssel für die Verschlüsselung verwenden, und einen anderen Schlüssel für die Entschlüsselung. Eine davon muss geheim gehalten werden, aber die andere kann öffentlich sein. Algorithmen, die öffentliche und private Schlüsselpaare verwenden, werden als *asymmetrisch* Algorithmen bezeichnet. Abbildung 2 auf Seite 11 veranschaulicht die asymmetrische Verschlüsselung von Schlüsseln, die auch als *Public-Key-Verschlüsselung* bezeichnet wird.

Die verwendeten Verschlüsselungs-und Entschlüsselungsalgorithmen können öffentlich sein, aber der Shared Secret-Schlüssel und der private Schlüssel müssen geheim gehalten werden.

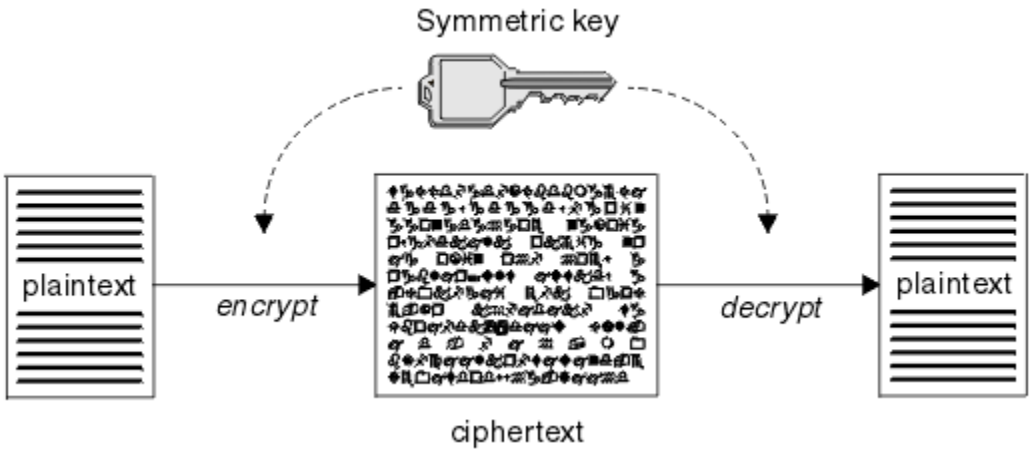


Abbildung 1. Symmetrische Schlüsselkryptografie

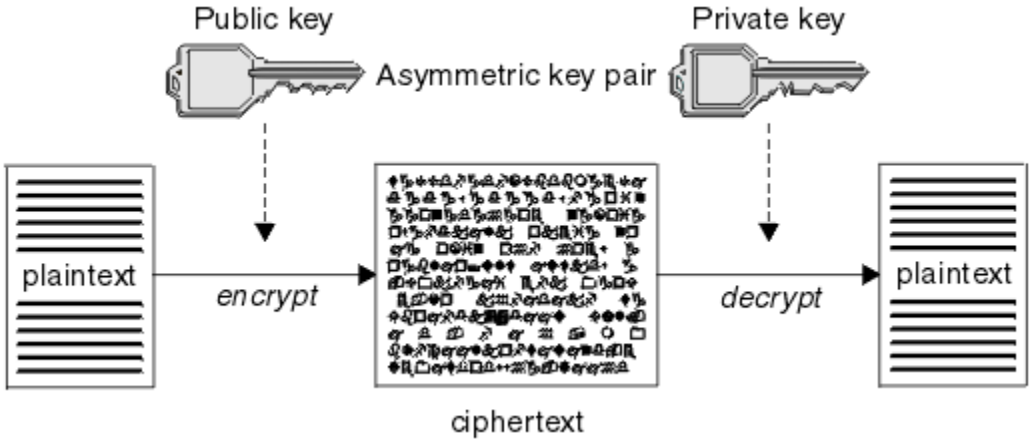


Abbildung 2. Asymmetrische Schlüsselkryptografie

Abbildung 2 auf Seite 11 zeigt unverschlüsselten Text, der mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und mit dem privaten Schlüssel des Empfängers entschlüsselt wird. Nur der vorgesehene Empfänger enthält den privaten Schlüssel zum Entschlüsseln des Chiffriktexts. Beachten Sie, dass der Sender auch Nachrichten mit einem privaten Schlüssel verschlüsseln kann, was es jedem erlaubt, den öffentlichen Schlüssel des Absenders zu entschlüsseln, um die Nachricht zu entschlüsseln, mit der Zusicherung, dass die Nachricht vom Absender gekommen sein muss.

Bei asymmetrischen Algorithmen werden Nachrichten entweder mit dem öffentlichen oder dem privaten Schlüssel verschlüsselt, können aber nur mit dem anderen Schlüssel entschlüsselt werden. Nur der private Schlüssel ist geheim, der öffentliche Schlüssel kann von jedem bekannt sein. Bei symmetrischen Algorithmen muss der gemeinsam genutzte Schlüssel nur den beiden Parteien bekannt sein. Dies wird als *Schlüsselverteilungsproblem* bezeichnet. Asymmetrische Algorithmen sind langsamer, haben aber den Vorteil, dass es kein Schlüsselverteilungsproblem gibt.

Weitere Terminologie, die der Kryptografie zugeordnet ist, ist:

Kraft
Die Stärke der Verschlüsselung wird durch die Schlüsselgröße bestimmt. Asymmetrische Algorithmen erfordern große Schlüssel, zum Beispiel:

1024 Bit	Asymmetrischer Schlüssel mit geringer Stärke
2048 Bit	Mittelstärkenasymmetrischer Schlüssel
4096 Bit	Hochfester asymmetrischer Schlüssel

Symmetrische Schlüssel sind kleiner: 256-Bit-Schlüssel geben Ihnen starke Verschlüsselung.

Blockchiffrierungsalgorithmus
Diese Algorithmen verschlüsseln Daten durch Blöcke. Der RC2-Algorithmus von RSA Data Security Inc. verwendet zum Beispiel Blöcke mit einer Länge von 8 Byte. Blockalgorithmen sind in der Regel langsamer als Datenstromalgorithmen.

Datenstromchiffrierungsalgorithmus
Diese Algorithmen arbeiten an jedem Byte an Daten. Datenstromalgorithmen sind in der Regel schneller als Blockalgorithmen.

Nachrichtendigests und digitale Signaturen

Ein Nachrichten-Digest ist eine numerische Darstellung des Inhalts einer Nachricht mit fester Größe. Der Nachrichten-Digest wird durch eine Hashfunktion berechnet und kann verschlüsselt werden, wobei eine digitale Signatur gebildet wird.

Die Hashfunktion, die zum Berechnen eines Nachrichten-Digest verwendet wird, muss zwei Kriterien erfüllen:

- Es muss ein Weg sein. Es darf nicht möglich sein, die Funktion umzukehren, um die Nachricht zu finden, die einem bestimmten Nachrichten-Digest entspricht, außer wenn alle möglichen Nachrichten getestet werden.
- Es muss rechenbar sein, zwei Nachrichten zu finden, die auf denselben Digest-Wert in Hash-Code-Datei (Hash) auftauchen.

Der Nachrichten-Digest wird mit der Nachricht selbst gesendet. Der Empfänger kann einen Digest für die Nachricht generieren und ihn mit dem Digest des Senders vergleichen. Die Integrität der Nachricht wird überprüft, wenn die beiden Nachrichtendigests identisch sind. Jede Manipulation der Nachricht während der Übertragung führt fast zu einem anderen Nachrichten-Digest.

Ein Nachrichten-Digest, der unter Verwendung eines geheimen symmetrischen Schlüssels erstellt wurde, wird als Nachrichtenauthentifizierungscode (Message Authentication Code, MAC) bezeichnet, da er die Zusicherung geben kann, dass die Nachricht nicht geändert wurde.

Der Sender kann auch einen Nachrichten-Digest generieren und dann den Digest mit Hilfe des privaten Schlüssels eines asymmetrischen Schlüsselpaares verschlüsseln und eine digitale Signatur bilden. Die Signatur muss dann vom Empfänger entschlüsselt werden, bevor sie mit einem lokal generierten Digest verglichen wird.

Zugehörige Konzepte

„Digitale Signaturen in SSL/TLS" auf Seite 22
Eine digitale Signatur wird gebildet, indem eine Darstellung einer Nachricht verschlüsselt wird. Die Verschlüsselung verwendet den privaten Schlüssel des Unterzeichners und arbeitet für die Effizienz in der Regel in der Regel in einem Nachrichten-Digest und nicht in der Nachricht selbst.

Digitale Zertifikate

Digitale Zertifikate schützen vor der Nachahmung; sie zertifizieren, dass ein öffentlicher Schlüssel zu einer bestimmten Entität gehört. Sie werden von einer Zertifizierungsstelle ausgegeben.

Digitale Zertifikate bieten Schutz vor der Aneignung, da ein digitales Zertifikat einen öffentlichen Schlüssel an seinen Eigner bindet, unabhängig davon, ob dieser Eigentümer eine Einzelperson, ein Warteschlangenmanager oder eine andere Entität ist. Digitale Zertifikate werden auch als öffentliche Schlüsselzertifikate bezeichnet, da sie Ihnen bei der Verwendung eines asymmetrischen Schlüsselschemas Zusicherungen über das Eigentumsrecht an einem öffentlichen Schlüssel geben. Ein digitales Zertifikat enthält den öffentlichen Schlüssel für eine Entität und ist eine Anweisung, die der öffentliche Schlüssel zu dieser Entität gehört:

- Wenn das Zertifikat für eine einzelne Entität vorhanden ist, wird das Zertifikat als *persönliches Zertifikat* oder *Benutzerzertifikat* bezeichnet.
- Wenn sich das Zertifikat für eine Zertifizierungsstelle befindet, wird das Zertifikat als *CA-Zertifikat* oder *Unterzeichnerzertifikat* bezeichnet.

Wenn öffentliche Schlüssel direkt von ihrem Eigner an eine andere Entität gesendet werden, besteht die Gefahr, dass die Nachricht abgefangen und der öffentliche Schlüssel durch einen anderen ersetzt wird. Dies wird als *Mann in der mittleren Attacke* bezeichnet. Die Lösung dieses Problems besteht darin, öffentliche Schlüssel über eine vertrauenswürdige dritte Partei auszutauschen und Ihnen eine sichere Zusicherung zu geben, dass der öffentliche Schlüssel wirklich zu der Entität gehört, mit der Sie kommunizieren. Anstatt den öffentlichen Schlüssel direkt zu senden, bitten Sie den vertrauenswürdigen Dritten, diese in ein digitales Zertifikat zu integrieren. Die vertrauenswürdige dritte Partei, die digitale Zertifikate ausgibt, wird als Zertifizierungsinstanz (CA) bezeichnet, wie in „Zertifizierungsstellen" auf Seite 13 beschrieben.

Was ist in einem digitalen Zertifikat?

Digitale Zertifikate enthalten bestimmte Informationen, die durch den X.509-Standard festgelegt sind.

Digitale Zertifikate, die von IBM MQ verwendet werden, sind mit dem X.509-Standard konform, der die erforderlichen Informationen und das entsprechende Sendeformat angibt. Dieser Standard definiert als Bestandteil der X.500-Standards die Rahmenbedingungen für die Authentifizierung.

Digitale Zertifikate enthalten mindestens die folgenden Informationen über die Entität, die zertifiziert wird:

- Der öffentliche Schlüssel des Eigners
- Der Registrierte Name des Eigners
- Den definierten Namen der CA, die das Zertifikat ausgestellt hat
- Das Datum, ab dem das Zertifikat gültig ist.
- Das Ablaufdatum des Zertifikats.
- Die Versionsnummer des Zertifikatsdatenformats, wie in X.509 definiert. Die aktuelle Version des X.509-Standards ist Version 3, und die meisten Zertifikate entsprechen dieser Version.
- Eine Seriennummer. Dies ist eine eindeutige Kennung, die von der Zertifizierungsstelle, die das Zertifikat ausgestellt hat, zugeordnet wurde. Die Seriennummer ist innerhalb der CA, die das Zertifikat ausgestellt hat, eindeutig: Es sind keine zwei Zertifikate vorhanden, die von demselben CA-Zertifikat signiert sind, die dieselbe Seriennummer haben.

Ein X.509-Zertifikat der Version 2 enthält außerdem eine Ausstellerkennung und eine Subjekt-ID, und ein X.509-Zertifikat der Version 3 kann eine Reihe von Erweiterungen enthalten. Einige Zertifikatserweiterungen, wie z. B. die Erweiterung "Basic Constraint", sind *standard* , andere sind jedoch implementierspezifisch. Eine Erweiterung kann *kritisch* sein. In diesem Fall muss ein System in der Lage sein, das Feld zu erkennen. Wenn es das Feld nicht erkennt, muss es das Zertifikat zurückweisen. Wenn eine Erweiterung nicht kritisch ist, kann das System sie ignorieren, wenn sie sie nicht erkennt.

Die digitale Signatur in einem persönlichen Zertifikat wird mit dem privaten Schlüssel der Zertifizierungsstelle generiert, die dieses Zertifikat signiert hat. Jeder, der das persönliche Zertifikat überprüfen muss, kann den öffentlichen Schlüssel der CA verwenden. Das CA-Zertifikat enthält seinen öffentlichen Schlüssel.

Digitale Zertifikate enthalten nicht Ihren privaten Schlüssel. Sie müssen Ihren geheimen Schlüssel geheim halten.

Anforderungen an persönliche Zertifikate
IBM MQ unterstützt digitale Zertifikate, die dem X.509-Standard entsprechen. Sie erfordert die Clientauthentifizierungsoption.

Da es sich bei IBM MQ um ein Peer-to-Peer-System handelt, wird es als Clientauthentifizierung in der SSL/TLS-Terminologie angesehen. Daher muss jedes persönliche Zertifikat, das für die SSL/TLS-Authentifizierung verwendet wird, eine Schlüsselverwendung der Clientauthentifizierung ermöglichen. Für nicht alle Serverzertifikate ist diese Option aktiviert, sodass der Zertifikatsprovider möglicherweise die Clientauthentifizierung auf der Stammzertifizierungsstelle für das sichere Zertifikat aktivieren muss.

Zusätzlich zu den Standards, die das Datenformat für ein digitales Zertifikat angeben, gibt es auch Standards für die Feststellung, ob ein Zertifikat gültig ist. Diese Standards wurden im Laufe der Zeit aktualisiert, um bestimmte Arten von Sicherheitsverletzungen zu verhindern. Beispiel: Ältere X.509-Zertifikate der Version 1 und 2 geben nicht an, ob das Zertifikat rechtmäßig zum Signieren anderer Zertifikate verwendet werden kann. Es war daher möglich, dass ein heimtückischer Benutzer ein persönliches Zertifikat aus einer legitimen Quelle erhält und neue Zertifikate erstellt, um andere Benutzer zu impersonieren.

Bei Verwendung von X.509-Zertifikaten der Version 3 werden die Zertifikatserweiterungen "BasicConstraints" und "KeyUsage" verwendet, um anzugeben, welche Zertifikate legitim andere Zertifikate signieren können. Der Standard IETF RFC 5280 gibt eine Reihe von Zertifikatvalidierungsregeln an, die die Anwendungssoftware implementieren muss, um Angriffsattacken zu verhindern. Eine Gruppe von Zertifikatregeln wird als Validierungsrichtlinie für Zertifikate bezeichnet.

Weitere Informationen zu Zertifikatsprüfrichtlinien finden Sie in IBM MQ finden Sie im Abschnitt „Zertifikatsprüfrichtlinien in IBM MQ” auf Seite 48.

Zertifizierungsstellen
Eine Zertifizierungsinstanz (CA) ist eine vertrauenswürdige dritte Partei, die digitale Zertifikate ausgibt, um Ihnen die Zusicherung zu geben, dass der öffentliche Schlüssel einer Entität wirklich zu dieser Entität gehört.

Die Rollen einer CA sind:

- Auf Anforderung eines digitalen Zertifikats, um die Identität des Anforderers vor dem Erstellen, Signieren und Zurückgeben des persönlichen Zertifikats zu überprüfen.
- Den eigenen öffentlichen Schlüssel der Zertifizierungsstelle in seinem CA-Zertifikat bereitstellen
- Listen von Zertifikaten veröffentlichen, die nicht mehr in einer Zertifikatswiderrufsliste (Certificate Revocation List, CRL) anerkannt sind. Weitere Informationen finden Sie in „Mit widerrufenden Zertifikaten arbeiten" auf Seite 362.
- Gehen Sie wie folgt vor, um den Zugriff auf den Widerrufstatus des Zertifikats durch den Betrieb eines OCSP-Responder

Definierte Namen
Der DN (Distinguished Name) identifiziert eine Entität in einem X.509-Zertifikat eindeutig.

Achtung: Es können nur die Attribute in der folgenden Tabelle in einem SSLPEER-Filter verwendet werden. Zertifikats-DNs können weitere Attribute enthalten, die Filterung nach diesen Attributen ist jedoch nicht zulässig.

Tabelle 1. Im DN vorkommende Attributtypen, die in einem SSLPEER-Filter verwendbar sind

Attributtyp	**Beschreibung**
SERIALANZAHL	Seriennummer des Zertifikats
MAIL	E-Mail-Adresse
E	E-Mail-Adresse (wird nicht weiter unterstützt; MAIL wird verwendet)
UID oder USERID	Benutzer-ID
CN	Allgemeiner Name
T	Titel
OU	Name der Organisationseinheit
Gleichstrom	Domänenkomponente
O	Name der Organisation
STREET	Straße / Erste Adresszeile
L	Lokalitätsname
ST (oder SP oder S)	Name des Bundeslandes oder der Provinz
PC	Postleitzahl
C	Land
UNSTRUKTUREDNAME	Hostname
UNSTRUKTUREDADRESSE	IP-Adresse
DNQ	Qualifikationsmerkmal für den definierten Namen

Der X.509-Standard definiert andere Attribute, die in der Regel nicht Teil des definierten Namens sind, aber optionale Erweiterungen für das digitale Zertifikat bereitstellen können.

Der X.509-Standard sieht vor, dass ein definierter Name in einem Zeichenfolgeformat angegeben wird. Beispiel:

```
CN=John Smith, OU=Test, O=IBM, C=GB
```

Der allgemeine Name (Common Name, CN) kann einen einzelnen Benutzer oder eine andere Entität beschreiben, z. B. einen Web-Server.

Der DN kann mehrere OU-und DC-Attribute enthalten. Es ist nur eine Instanz jedes der anderen Attribute zulässig. Die Reihenfolge der OU-Einträge ist von Bedeutung: Die Reihenfolge gibt eine Hierarchie der Organisationseinheitennamen an, wobei die höchste Ebene zuerst die Ebene der höchsten Ebene enthält. Die Reihenfolge der DC-Einträge ist ebenfalls signifikant.

IBM MQ toleriert bestimmte fehlerhafte definierte Namen. Weitere Informationen finden Sie unter IBM MQ-Regeln für SSLPEER-Werte.

Zugehörige Konzepte

„Was ist in einem digitalen Zertifikat?" auf Seite 12
Digitale Zertifikate enthalten bestimmte Informationen, die durch den X.509-Standard festgelegt sind.

Persönliche Zertifikate von einer Zertifizierungsstelle anfordern

Sie können ein Zertifikat von einer anerkannten externen Zertifizierungsstelle (CA) anfordern.

Sie erhalten ein digitales Zertifikat, indem Sie Informationen an eine CA senden, in Form einer Zertifikatsanforderung. Der X.509-Standard definiert ein Format für diese Informationen, aber einige CAs haben ein eigenes Format. Zertifikatsanforderungen werden in der Regel von dem Zertifikatsmanagementtool generiert, das vom System verwendet wird. Beispiel:

- Multi Der Befehl **strmqikm** (iKeyman-Tool) in Multiplatforms und die Befehle **runmqckm** sowie **runmqakm** in AIX, Linux®, and Windows.
- z/OS RACF unter z/OS.

Die Informationen enthalten den definierten Namen (DN) und den öffentlichen Schlüssel. Wenn Ihr Zertifikat-Management-Tool Ihre Zertifikatsanforderung generiert, generiert es auch Ihren privaten Schlüssel, den Sie sicher behalten müssen. Verteilen Sie niemals Ihren privaten Schlüssel.

Wenn die CA Ihre Anfrage erhält, verifiziert die Behörde Ihre Identität, bevor sie das Zertifikat erstellt und sie als persönliches Zertifikat an Sie zurückgibt.

Abbildung 3 auf Seite 15 veranschaulicht den Prozess, mit dem ein digitales Zertifikat von einer Zertifizierungsstelle abgerufen wird.

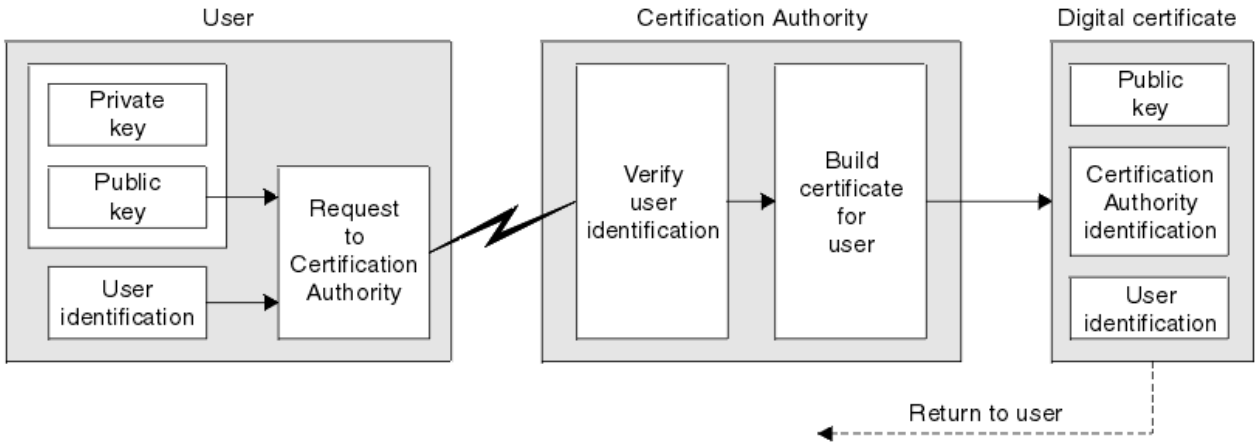


Abbildung 3. Abrufen eines digitalen Zertifikats

Im Diagramm:

- Die Benutzeridentifikation enthält den definierten Namen des Subjekts.
- Die ID der Zertifizierungsstelle enthält den definierten Namen der CA, die das Zertifikat ausgestellt hat.

Digitale Zertifikate enthalten zusätzliche Felder, die nicht im Diagramm dargestellt sind. Weitere Informationen zu den anderen Feldern in einem digitalen Zertifikat finden Sie in „Was ist in einem digitalen Zertifikat?" auf Seite 12.

Funktionsweise der Zertifikatsketten
Wenn Sie das Zertifikat für eine andere Entität empfangen, müssen Sie unter Umständen eine *Zertifikatskette* verwenden, um das Zertifikat *root CA* zu erhalten.

Die Zertifikatskette, die auch als *Zertifizierungspfad* bezeichnet wird, ist eine Liste der Zertifikate, die für die Authentifizierung einer Entität verwendet werden. Die Kette oder der Pfad beginnt mit dem Zertifikat dieser Entität, und jedes Zertifikat in der Kette wird von der Entität signiert, die durch das nächste Zertifikat in der Kette identifiziert wird. Die Kette wird mit einem Root-CA-Zertifikat beendet. Das Stammzertifikat der Zertifizierungsstelle wird immer von der Zertifizierungsstelle (CA) selbst signiert. Die Signaturen aller Zertifikate in der Kette müssen bis zum Zertifikat der Stammzertifizierungsstelle überprüft und bestätigt werden.

Abbildung 4 auf Seite 16 zeigt einen Zertifizierungspfad vom Zertifikateigner bis zur Stamm-CA, wo die Vertrauenskette beginnt.

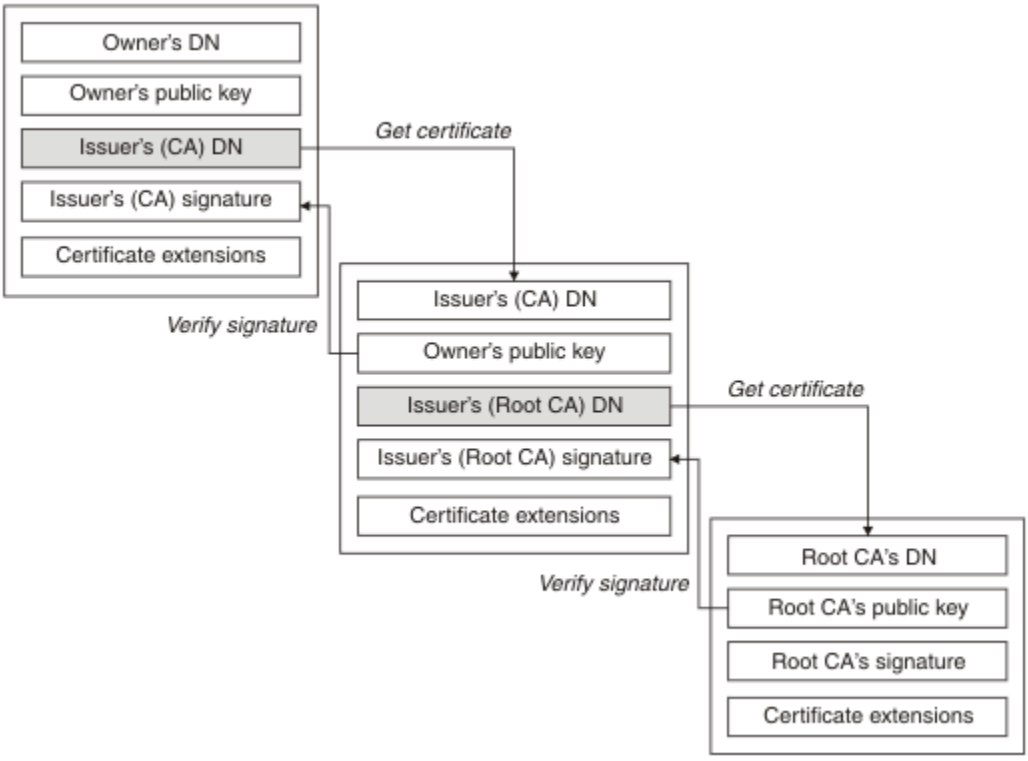


Abbildung 4. Kette der Vertrauenskette

Jedes Zertifikat kann eine oder mehrere Erweiterungen enthalten. Ein Zertifikat, das zu einer Zertifizierungsstelle gehört, enthält in der Regel eine Erweiterung "BasicConstraints" mit dem Flag "isCA", um anzuzeigen, dass es zulässig ist, andere Zertifikate zu signieren.

Wenn Zertifikate nicht mehr gültig sind
Digitale Zertifikate können ablaufen oder widerrufen werden.

Digitale Zertifikate werden für einen bestimmten Zeitraum ausgestellt, nach dessen Ablauf sie nicht mehr gültig sind.

Zertifikate können aus verschiedenen Gründen widerrufen werden, z. B.:

- Der Eigner wurde in eine andere Organisation verschoben.
- Der private Schlüssel ist nicht mehr geheim.

IBM MQ kann überprüfen, ob ein Zertifikat widerrufen wurde, indem eine Anforderung an den OCSP-Responder (Online Certificate Status Protocol) gesendet wird (nur unter AIX, Linux, and Windows). Alternativ können sie auf eine Zertifikatswiderrufsliste (Certificate Revocation List, CRL) auf einem LDAP-Server zu-

greifen. Die OCSP-Widerrufs-und CRL-Informationen werden von einer Zertifizierungsstelle veröffentlicht. Weitere Informationen finden Sie im Abschnitt „Mit widerrufenden Zertifikaten arbeiten” auf Seite 362.

Public Key Infrastructure (PKI)

Eine PKI (Public Key Infrastructure) ist ein System von Einrichtungen, Richtlinien und Services, die die Verwendung öffentlicher Verschlüsselungsschlüssel für die Authentifizierung der an einer Transaktion beteiligten Parteien unterstützt.

Es gibt keinen einzigen Standard, der die Komponenten einer Public-Key-Infrastruktur definiert, aber eine PKI umfasst normalerweise Zertifizierungsstellen (CAs) und Registrierungsberechtigungen (RAs). CAs stellen die folgenden Services bereit:

- Digitale Zertifikate ausstellen
- Digitale Zertifikate validieren
- Digitale Zertifikate werden zurückgeschworen
- Öffentliche Schlüssel verteilen

Die X.509-Standards bilden die Basis für die Industrie-Standard Public Key Infrastructure.

Weitere Informationen zu digitalen Zertifikaten und Zertifizierungsstellen (CAs) finden Sie im Abschnitt „Digitale Zertifikate” auf Seite 12. RAs überprüfen die Informationen, die bereitgestellt werden, wenn digitale Zertifikate angefordert werden. Prüft der RA diese Informationen, kann die Zertifizierungsstelle ein digitales Zertifikat an den Anforderer ausgeben.

Eine PKI kann auch Tools zum Verwalten von digitalen Zertifikaten und öffentlichen Schlüsseln bereitstellen. Eine PKI wird manchmal auch als *Vertrauenshierarchie* für die Verwaltung digitaler Zertifikate beschrieben, aber die meisten Definitionen enthalten zusätzliche Services. Einige Definitionen umfassen Verschlüsselungs-und digitale Signaturservices, aber diese Services sind für den Betrieb einer PKI nicht unbedingt erforderlich.

Verschlüsselte Sicherheitsprotokolle: TLS

Verschlüsselte Protokolle stellen sichere Verbindungen bereit, die es zwei Parteien ermöglichen, mit Datenschutz und Datenintegrität zu kommunizieren. Das TLS-Protokoll (Transport Layer Security) wurde von dem Protokoll Secure Sockets Layer (SSL) entwickelt. IBM MQ uunterstützt TLS.

Die primären Ziele beider Protokolle sind die Gewährleistung der Vertraulichkeit (manchmal auch als *Datenschutz* bezeichnet), die Datenintegrität, die Identifikation und die Authentifizierung mit Hilfe digitaler Zertifikate.

Obwohl beide Protokolle ähnlich sind, sind die Unterschiede doch so gravierend, dass SSL 3.0 und die verschiedenen TLS-Versionen funktionell nicht aufeinander abgestimmt sind.

Zugehörige Konzepte

„TLS-Sicherheitsprotokolle in IBM MQ” auf Seite 26
IBM MQ unterstützt das TLS-Protokoll (Transport Layer Security), um die Sicherheit auf Verbindungsebene für Nachrichtenkanäle und MQI-Kanäle bereitzustellen.

Konzepte der Transport Layer Security (TLS)

Das TLS-Protokoll ermöglicht es zwei Parteien, sich gegenseitig zu identifizieren und zu authentifizieren und mit Vertraulichkeit und Datenintegrität zu kommunizieren. Das TLS-Protokoll wurde vom Netscape SSL 3.0-Protokoll entwickelt, aber TLS und SSL sind nicht interaktiv.

Das TLS-Protokoll ermöglicht Kommunikationssicherheit im Internet sowie die vertrauliche und zuverlässige Kommunikation zwischen Client/Server-Anwendungen. Die Protokolle bestehen aus zwei Schichten: einem Record Protocol und einem Handshake Protocol, die über ein Transportprotokoll wie TCP/IP geschichtet sind. Sie verwenden sowohl asymmetrische als auch symmetrische Kryptographietechniken.

Eine TLS-Verbindung wird von einer Anwendung initiiert, die zum TLS-Client wird. Die Anwendung, die die Verbindung empfängt, wird zum TLS-Server. Jede neue Sitzung beginnt mit einem Handshake, wie er durch die TLS-Protokolle definiert wird.

Eine vollständige Liste der von IBM MQ unterstützten CipherSpecs finden Sie unter „CipherSpecs aktivieren” auf Seite 448.

Weitere Informationen zum SSL-Protokoll finden Sie in den Informationen unter https://developer.mozilla.org/docs/Mozilla/Projects/NSS. Weitere Informationen zum TLS-Protokoll werden von der TLS Working Group auf der Website der Internet Engineering Task Force unter https://www.ietf.org bereitgestellt.

Überblick über den SSL/TLS-Handshake

Der SSL/TLS-Handshake ermöglicht dem TLS-Client und dem TLS-Server, die geheimen Schlüssel zu erstellen, mit denen sie kommunizieren.

Dieser Abschnitt enthält eine Zusammenfassung der Schritte, mit denen der TLS-Client und der TLS-Server miteinander kommunizieren können.

- Akzeptieren Sie die Version des zu verwendenden Protokolls.
- Chiffrieralgorithmen auswählen.
- sich gegenseitig über den Austausch und die Überprüfung digitaler Zertifikate authentifizieren
- Verwenden Sie asymmetrische Verschlüsselungsverfahren, um einen gemeinsamen geheimen Schlüssel zu generieren, der das Hauptverteilungsproblem vermeidet. TLS verwendet dann den gemeinsam genutzten Schlüssel für die symmetrische Verschlüsselung von Nachrichten, die schneller als asymmetrische Verschlüsselung ist.

Weitere Informationen zu kryptografischen Algorithmen und digitalen Zertifikaten finden Sie in den zugehörigen Informationen.

In der Übersicht sind die Schritte im TLS-Handshake wie folgt:

1. Der TLS-Client sendet eine " Client-Hello " -Nachricht, in der Verschlüsselungsdaten wie die TLS-Version und in der Reihenfolge der Vorgaben des Clients die vom Client unterstützten CipherSuites aufgelistet werden. Die Nachricht enthält auch eine zufällige Bytefolge, die in nachfolgenden Berechnungen verwendet wird. Das Protokoll ermöglicht es dem " Clienthello " , die vom Client unterstützten Datenkomprimierungsmethoden einzuschließen.
2. Der TLS-Server antwortet mit einer Nachricht vom Typ " server hello " , die die CipherSuite enthält, die vom Server aus der vom Client bereitgestellten Liste, der Sitzungs-ID und einer anderen wahlfreien Bytefolge ausgewählt wurde. Der Server sendet auch sein digitales Zertifikat. Wenn für den Server ein digitales Zertifikat für die Clientauthentifizierung erforderlich ist, sendet der Server eine " Clientzertifikatsanforderung " , die eine Liste der unterstützten Typen von Zertifikaten und die definierten Namen akzeptabler Zertifizierungsstellen (CAs) enthält.
3. Der TLS-Client überprüft das digitale Zertifikat des Servers. Weitere Informationen finden Sie unter „Wie TLS Identifikation, Authentifizierung, Vertraulichkeit und Integrität bereitstellt” auf Seite 19.
4. Der TLS-Client sendet die zufällige Bytefolge, die es sowohl dem Client als auch dem Server ermöglicht, den geheimen Schlüssel zu berechnen, der für die Verschlüsselung der nachfolgenden Nachrichtendaten verwendet werden soll. Die zufällige Bytefolge selbst wird mit dem öffentlichen Schlüssel des Servers verschlüsselt.
5. Wenn der TLS-Server eine " Clientzertifikatsanforderung " gesendet hat, sendet der Client eine zufällige Bytefolge, die mit dem privaten Schlüssel des Clients verschlüsselt wird, zusammen mit dem digitalen Zertifikat des Clients oder mit einem " Alert für kein digitales Zertifikat ". Dieser Alert ist nur eine Warnung, aber bei einigen Implementierungen schlägt der Handshake fehl, wenn die Clientauthentifizierung obligatorisch ist.
6. Der TLS-Server überprüft das Clientzertifikat. Weitere Informationen finden Sie unter „Wie TLS Identifikation, Authentifizierung, Vertraulichkeit und Integrität bereitstellt” auf Seite 19.
7. Der TLS-Client sendet dem Server eine " fertige " Nachricht, die mit dem geheimen Schlüssel verschlüsselt wird, was darauf hinweist, dass der Client Teil des Handshake abgeschlossen ist.
8. Der TLS-Server sendet dem Client eine " fertige " Nachricht, die mit dem geheimen Schlüssel verschlüsselt wird, was darauf hinweist, dass der Serverteil des Handshake abgeschlossen ist.
9. Für die Dauer der TLS-Sitzung kann der Server und Client jetzt Nachrichten austauschen, die symmetrisch mit dem geheimen Schlüssel für gemeinsame Nutzung verschlüsselt sind.

Abbildung 5 auf Seite 19 veranschaulicht den TLS-Handshake.

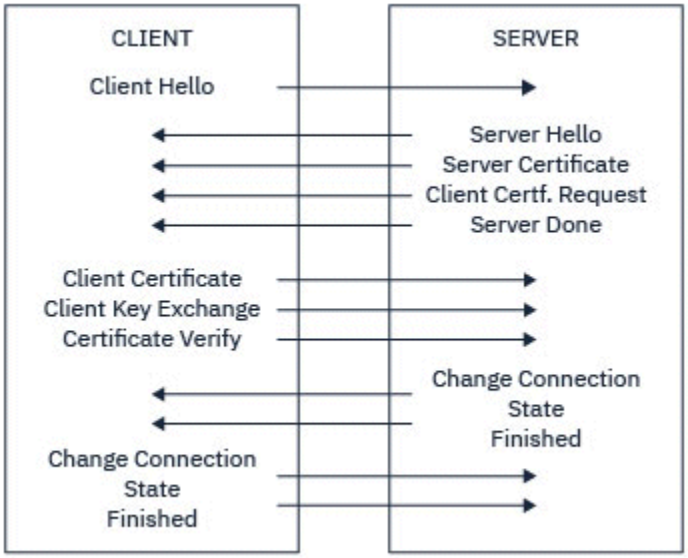


Abbildung 5. Übersicht über den TLS-Handshake

Wie TLS Identifikation, Authentifizierung, Vertraulichkeit und Integrität bereitstellt

Während der Client-und Serverauthentifizierung ist ein Schritt erforderlich, der Daten mit einem der Schlüssel in einem asymmetrischen Schlüsselpaar verschlüsselt und mit dem anderen Schlüssel des Paares entschlüsselt. Es wird ein Nachrichten-Digest verwendet, um die Integrität zu gewährleisten.

Eine Übersicht über die Schritte im Zusammenhang mit dem TLS-Handshake finden Sie unter „Überblick über den SSL/TLS-Handshake” auf Seite 18.

Authentifizierung durch TLS

Für die Serverauthentifizierung verwendet der Client den öffentlichen Schlüssel des Servers, um die Daten zu verschlüsseln, die zur Berechnung des geheimen Schlüssels verwendet werden. Der Server kann den geheimen Schlüssel nur generieren, wenn er diese Daten mit dem richtigen privaten Schlüssel entschlüsseln kann. Die zufällige Bytefolge selbst wird mit dem öffentlichen Schlüssel des Servers verschlüsselt (Schritt „4” auf Seite 18 in der Übersicht).

Für die Clientauthentifizierung verwendet der Server den öffentlichen Schlüssel im Clientzertifikat, um die Daten zu entschlüsseln, die der Client während des Schritts „5” auf Seite 18 des Handshake sendet. Der Austausch von fertig gestellten Nachrichten, die mit dem geheimen Schlüssel verschlüsselt sind (Schritte „7” auf Seite 18 und „8” auf Seite 18 in der Übersicht), bestätigt, dass die Authentifizierung abgeschlossen ist.

Wenn einer der Authentifizierungsschritte fehlschlägt, schlägt der Handshake fehl, und die Sitzung wird beendet.

Der Austausch digitaler Zertifikate während des TLS-Handshake ist Teil des Authentifizierungsprozesses. Weitere Informationen darüber, wie Zertifikate Schutz vor der Ipersonation bieten, finden Sie in den zugehörigen Informationen. Die erforderlichen Zertifikate sind wie folgt, wobei CA X das Zertifikat an den TLS-Client ausgibt und CA Y das Zertifikat auf den TLS-Server setzt:

Nur für die Serverauthentifizierung benötigt der TLS-Server:

- Das persönliche Zertifikat, das dem Server von Zertifizierungsstelle Y ausgestellt wurde.
- Der private Schlüssel des Servers

und die TLS-Clientanforderungen:

- Das CA-Zertifikat für Zertifizierungsstelle Y

Wenn der TLS-Server eine Clientauthentifizierung erfordert, überprüft der Server die Identität des Clients, indem er das digitale Zertifikat des Clients mit dem öffentlichen Schlüssel für die Zertifizierungsstelle überprüft, die das persönliche Zertifikat für den Client ausgestellt hat (in diesem Fall CA X). Für die Server-und Clientauthentifizierung benötigt der Server Folgendes:

- Das persönliche Zertifikat, das dem Server von Zertifizierungsstelle Y ausgestellt wurde.
- Der private Schlüssel des Servers
- Das CA-Zertifikat für Zertifizierungsstelle X

und der Client benötigt:

- Das persönliche Zertifikat, das dem Client von Zertifizierungsstelle X ausgegeben wurde
- Der private Schlüssel des Clients
- Das CA-Zertifikat für Zertifizierungsstelle Y

Sowohl der TLS-Server als auch der Client benötigen möglicherweise andere CA-Zertifikate, um eine Zertifikatskette zum Stamm-CA-Zertifikat zu bilden. Weitere Informationen zu Zertifikatsketten finden Sie in den zugehörigen Informationen.

Was während der Zertifikatsprüfung passiert

Wie in den Schritten „3" auf Seite 18 und „6" auf Seite 18 der Übersicht angegeben, überprüft der TLS-Client das Serverzertifikat und der TLS-Server prüft das Zertifikat des Kunden. Für diese Überprüfung gibt es vier Aspekte:

1. Die digitale Signatur wird geprüft (siehe „Digitale Signaturen in SSL/TLS" auf Seite 22).
2. Die Zertifikatskette wird geprüft. Sie sollten über temporäre CA-Zertifikate verfügen (siehe „Funktionsweise der Zertifikatsketten" auf Seite 16).
3. Das Verfallsdatum und die Gültigkeitsdauer werden überprüft.
4. Der Widerrufsstatus des Zertifikats wird überprüft (siehe „Mit widerrufenden Zertifikaten arbeiten" auf Seite 362).

Geheimer Schlüssel zurückgesetzt

Während eines TLS-Handshake wird ein *geheimer Schlüssel* generiert, um Daten zwischen dem TLS-Client und dem TLS-Server zu verschlüsseln. Der geheime Schlüssel wird in einer mathematischen Formel verwendet, die auf die Daten angewendet wird, um Klartext in nicht lesbaren Chiffriertext umzuwandeln, und ciphertext in unverschlüsselbaren Text.

Der geheime Schlüssel wird aus dem wahlfreien Text generiert, der als Teil des Handshake gesendet wird, und wird zum Verschlüsseln von Klartext in Chiffriertext verwendet. Der geheime Schlüssel wird auch im MAC-Algorithmus (Message Authentication Code) verwendet, der verwendet wird, um festzustellen, ob eine Nachricht geändert wurde. Weitere Informationen finden Sie unter „Nachrichtendigests und digitale Signaturen" auf Seite 11.

Wenn der geheime Schlüssel erkannt wird, kann der unverschlüsselte Text einer Nachricht aus dem Chiffriertext entschlüsselt werden, oder der Nachrichtendigest kann berechnet werden, so dass Nachrichten ohne Erkennung geändert werden können. Selbst bei einem komplexen Algorithmus kann der Klartext ermittelt werden, indem jede mögliche mathematische Transformation auf den Chiffriertext angewendet wird. Um die Menge der Daten, die entschlüsselt oder geändert werden können, zu minimieren, wenn der geheime Schlüssel beschädigt ist, kann der geheime Schlüssel in regelmäßigen Abständen neu vereinbart werden. Wenn der geheime Schlüssel neu verhandelt wurde, kann der vorherige geheime Schlüssel nicht mehr verwendet werden, um Daten zu entschlüsseln, die mit dem neuen geheimen Schlüssel verschlüsselt wurden.

Wie TLS Vertraulichkeit gewährleistet

TLS verwendet eine Kombination aus symmetrischer und asymmetrischer Verschlüsselung, um die Vertraulichkeit von Nachrichten zu gewährleisten. Während des TLS-Handshake stimmen der TLS-Client und

der TLS-Client einen Verschlüsselungsalgorithmus und einen gemeinsam genutzten geheimen Schlüssel nur für eine Sitzung zu. Alle Nachrichten, die zwischen dem TLS-Client und dem TLS-Server übertragen werden, werden mit diesem Algorithmus und Schlüssel verschlüsselt, wobei sichergestellt wird, dass die Nachricht auch dann privat bleibt, wenn sie abgefangen wird. Da TLS bei der Übertragung des Shared Secret-Schlüssels asymmetrische Verschlüsselung verwendet, gibt es kein Problem mit der Schlüsselverteilung. Weitere Informationen zu Verschlüsselungsverfahren finden Sie in „Kryptografie" auf Seite 10.

Integrität von TLS

TLS bietet Datenintegrität durch die Berechnung eines Nachrichten-Digest. Weitere Informationen finden Sie in „Datenintegrität von Nachrichten" auf Seite 509.

Die Verwendung von TLS stellt die Datenintegrität sicher, vorausgesetzt, die CipherSpec in Ihrer Kanaldefinition verwendet einen Hashalgorithmus, wie in der Tabelle in „CipherSpecs aktivieren" auf Seite 448 beschrieben.

Wenn die Datenintegrität ein Problem ist, sollten Sie vermeiden, eine CipherSpec auszuwählen, deren Hashalgorithmus als "None" ("None") aufgeführt ist. Die Verwendung von MD5 wird auch stark entmutert, da dies jetzt sehr alt und für die meisten praktischen Zwecke nicht mehr sicher ist.

CipherSpecs und CipherSuites

Kryptografische Sicherheitsprotokolle müssen sich auf die Algorithmen einigen, die von einer sicheren Verbindung verwendet werden. CipherSpecs und CipherSuites definieren bestimmte Kombinationen von Algorithmen.

Eine CipherSpec identifiziert eine Kombination aus Verschlüsselungsalgorithmus und Algorithmus für Nachrichtenauthentifizierungscode (MAC). Beide Enden einer TLS-Verbindung müssen sich auf dieselbe CipherSpec einigen, um kommunizieren zu können.

IBM MQ unterstützt TLS1.3-und TLS1.2-Protokolle und CipherSpecs. Sie können jedoch veraltete CipherSpecs aktivieren, wenn dies erforderlich ist.

Weitere Informationen finden Sie unter „CipherSpecs aktivieren" auf Seite 448:

- CipherSpecs, die von IBM MQ unterstützt werden
- Veraltete CipherSpecs für SSL 3.0 und TLS 1.0 aktivieren

Wichtig: Bei der Bearbeitung von IBM MQ-Kanälen verwenden Sie eine CipherSpec. Bei der Bearbeitung von Java-Kanälen, JMS-Kanälen oder MQTT-Kanälen können Sie eine CipherSuite angeben.

Weitere Informationen über CipherSpecs finden Sie unter „CipherSpecs aktivieren" auf Seite 448.

Eine CipherSuite ist eine Suite von Verschlüsselungsalgorithmen, die von einer TLS-Verbindung verwendet werden. Eine Suite besteht aus drei unterschiedlichen Algorithmen:

- Der Schlüsselaustausch-und Authentifizierungsalgorithmus, der während des Handshake verwendet wird
- Der Verschlüsselungsalgorithmus, der zum Verschlüsseln der Daten verwendet wird.
- Der MAC-Algorithmus (Message Authentication Code), der zum Generieren des Nachrichten-Digest verwendet wird.

Es gibt mehrere Optionen für jede Komponente der Suite, aber nur bestimmte Kombinationen sind gültig, wenn sie für eine TLS-Verbindung angegeben werden. Der Name einer gültigen CipherSuite definiert die Kombination der verwendeten Algorithmen. Die CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA gibt z. B. Folgendes an:

- Der Algorithmus für RSA-Schlüsselaustausch und -Authentifizierung
- Der AES-Verschlüsselungsalgorithmus unter Verwendung eines 128-Bit-Schlüssels und dem Modus zur Verkettung von Verschlüsselungsblöcken (CBC)
- Der SHA-1-Nachrichtenauthentifizierungscode (MAC)

Digitale Signaturen in SSL/TLS

Eine digitale Signatur wird gebildet, indem eine Darstellung einer Nachricht verschlüsselt wird. Die Verschlüsselung verwendet den privaten Schlüssel des Unterzeichners und arbeitet für die Effizienz in der Regel in der Regel in einem Nachrichten-Digest und nicht in der Nachricht selbst.

Digitale Signaturen variieren mit den Daten, die signiert werden, im Gegensatz zu handgeschriebenen Signaturen, die nicht vom Inhalt des signierten Dokuments abhängen. Wenn zwei verschiedene Nachrichten von derselben Entität digital signiert werden, unterscheiden sich die beiden Signaturen voneinander, aber beide Signaturen können mit demselben öffentlichen Schlüssel verifiziert werden, d.

Die Schritte des digitalen Signaturprozesses sind wie folgt:

1. Der Sender berechnet einen Nachrichten-Digest und verschlüsselt dann den Digest mit dem privaten Schlüssel des Absenders, der die digitale Signatur bildet.
2. Der Sender überträgt die digitale Signatur mit der Nachricht.
3. Der Empfänger entschlüsselt die digitale Signatur mit dem öffentlichen Schlüssel des Absenders und regeneriert den Nachrichtendigest des Absenders.
4. Der Empfänger berechnet einen Nachrichten-Digest aus den empfangenen Nachrichtendaten und verifiziert, dass die beiden Digests identisch sind.

Abbildung 6 auf Seite 22 veranschaulicht diesen Prozess.

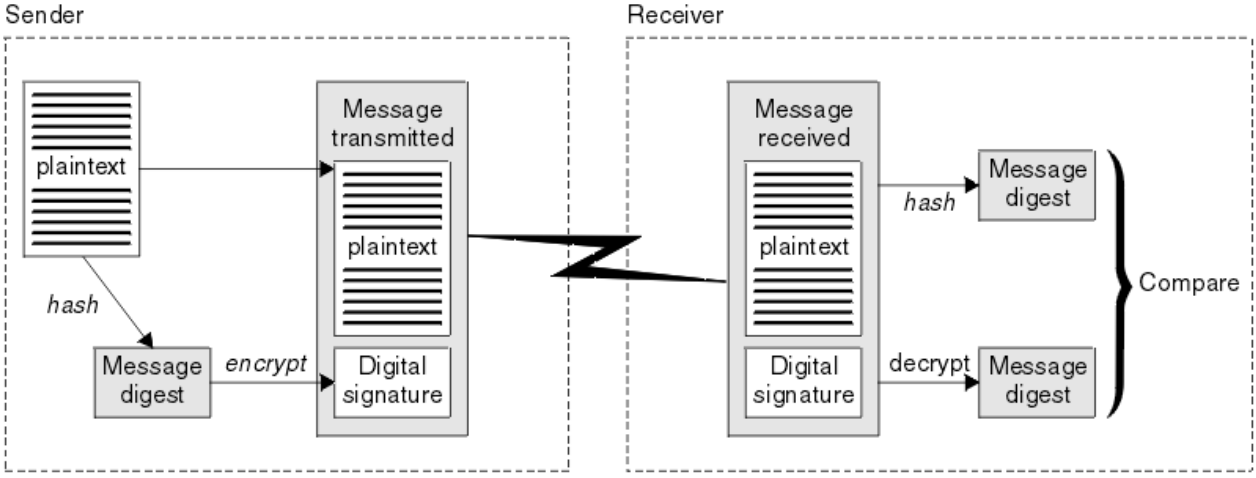


Abbildung 6. Der Prozess der digitalen Signatur

Wenn die digitale Signatur geprüft wird, weiß der Empfänger, dass:

- Die Nachricht wurde während der Übertragung nicht geändert.
- Die Nachricht wurde von der Entität gesendet, die behauptet hat, sie gesendet zu haben.

Digitale Signaturen sind Teil der Integritäts-und Authentifizierungsservices. Digitale Signaturen stellen auch Ursprungsnachweise zur Verfügung. Nur der Absender kennt den privaten Schlüssel, der einen starken Beweis dafür liefert, dass der Absender der Absender der Nachricht ist.

Anmerkung: Sie können auch die Nachricht selbst verschlüsseln, die die Vertraulichkeit der Informationen in der Nachricht schützt.

Federal Information Processing Standards

Die US-Regierung produziert technische Beratung zu IT-Systemen und Sicherheit, einschließlich der Datenverschlüsselung. Das National Institute for Standards and Technology (NIST) ist ein wichtiges Gremium, das sich mit IT-Systemen und der Sicherheit befasst. NIST erstellt Empfehlungen und Standards, einschließlich der Federal Information Processing Standards (FIPS).

Ein signifikanter dieser Standards ist FIPS 140-2, was die Verwendung von starken kryptografischen Algorithmen erfordert. FIPS 140-2 gibt außerdem Anforderungen an Hashing-Algorithmen an, die zum Schutz von Paketen vor Änderungen im Transit verwendet werden sollen.

Anmerkung: Unter AIX, Linux, and Windowsstellt IBM MQ Konformität mit FIPS 140-2 über das Verschlüsselungsmodul "IBM Crypto for C" bereit. Das Zertifikat für dieses Modul wurde in den Langzeitstatus versetzt. Kunden sollten das IBM Crypto for C-Zertifikat anzeigen und sich über Empfehlungen von NIST im Klaren sein. Ein Ersatz-FIPS 140-3-Modul ist derzeit in Bearbeitung und sein Status kann angezeigt werden, indem in der NIST-CMVP-Module in der Prozesslistenach ihm gesucht wird.

IBM MQ stellt die FIPS 140-2-Unterstützung bereit, wenn die Konfiguration entsprechend vorgenommen wurde.

Im Laufe der Zeit entwickeln Analysten Angriffe auf vorhandene Verschlüsselungs-und Hash-Algorithmen. Es werden neue Algorithmen angenommen, um diesen Angriffen zu widerstehen. FIPS 140-2 wird in regelmäßigen Abständen aktualisiert, um diesen Änderungen Rechnung zu tragen.

Zugehörige Konzepte

„National Security Agency (NSA) Suite B Cryptography" auf Seite 23
Die Regierung der Vereinigten Staaten von Amerika erstellt technische Beratung zu IT-Systemen und Sicherheit, einschließlich der Datenverschlüsselung. Die US National Security Agency (NSA) empfiehlt eine Reihe interoperabler kryptographischer Algorithmen in ihrem Suite B Standard.

National Security Agency (NSA) Suite B Cryptography

Die Regierung der Vereinigten Staaten von Amerika erstellt technische Beratung zu IT-Systemen und Sicherheit, einschließlich der Datenverschlüsselung. Die US National Security Agency (NSA) empfiehlt eine Reihe interoperabler kryptographischer Algorithmen in ihrem Suite B Standard.

Der Suite B-Standard gibt einen Betriebsmodus an, in dem nur eine bestimmte Gruppe von sicheren Verschlüsselungsalgorithmen verwendet wird. Der Standard Suite B gibt Folgendes an:

- Verschlüsselungsalgorithmus (AES)
- Der Schlüsselaustauschalgorithmus (Elliptic Curve Diffie-Hellman, auch bekannt als ECDH)
- Algorithmus für digitale Signatur (Elliptic Curve Digital Signature Algorithm, auch bekannt als ECDSA)
- Die Hashing-Algorithmen (SHA-256 oder SHA-384)

Darüber hinaus gibt der IETF-Standard RFC 6460 Suite B-konforme Profile an, die die detaillierte Anwendungskonfiguration und das erforderliche Verhalten definieren, die erforderlich sind, um den Standard-Suite B-Standards einzuhalten. Es definiert zwei Profile:

1. Ein Suite B-konformes Profil für die Verwendung mit TLS 1.2. Bei der Konfiguration für eine Suite B-konforme Operation wird nur die eingeschränkte Gruppe von Verschlüsselungsalgorithmen verwendet.
2. Ein Übergangsprofil für die Verwendung mit TLS 1.0 oder TLS 1.1. Dieses Profil ermöglicht die Interoperabilität mit nicht-Suite B-kompatiblen Servern. Bei der Konfiguration für die Suite B-Übergangsoperation können zusätzliche Verschlüsselungs-und Hash-Algorithmen verwendet werden.

Der Suite B-Standard ist konzeptionell ähnlich wie FIPS 140-2, da er die Menge aktivierter kryptografischer Algorithmen einschränkt, um ein gesichertes Sicherheitsniveau zu gewährleisten.

Auf AIX, Linux, and Windows-Systemen kann IBM MQ so konfiguriert werden, dass es mit dem Suite B-konformen TLS 1.2-Profil konform ist, aber das Suite B-Übergangsprofil nicht unterstützt. Weitere Informationen finden Sie in „NSA Suite B-Verschlüsselung in IBM MQ" auf Seite 45.

Zugehörige Verweise

„Federal Information Processing Standards" auf Seite 22
Die US-Regierung produziert technische Beratung zu IT-Systemen und Sicherheit, einschließlich der Datenverschlüsselung. Das National Institute for Standards and Technology (NIST) ist ein wichtiges Gremium, das sich mit IT-Systemen und der Sicherheit befasst. NIST erstellt Empfehlungen und Standards, einschließlich der Federal Information Processing Standards (FIPS).

IBM MQ-Sicherheitsmechanismen

In dieser Themensammlung finden Sie Informationen zum Implementieren der verschiedenen Sicherheitskonzepte in IBM MQ.

IBM MQ stellt Mechanismen bereit, mit denen alle in „Sicherheitskonzepte und -mechanismen” auf Seite 7 eingeführten Sicherheitskonzepte implementiert werden. Diese werden in den folgenden Abschnitten ausführlicher behandelt.

Identifikation und Authentifizierung in IBM MQ

In IBM MQ können Sie die Identifikation und Authentifizierung mithilfe von Informationen zum Nachrichtenkontext und einer gegenseitigen Authentifizierung implementieren.

Im Folgenden finden Sie einige Beispiele für die Identifikation und Authentifizierung in einer IBM MQ-Umgebung:

- Jede Nachricht kann *Nachrichtenkontext* -Informationen enthalten. Diese Informationen werden im Nachrichtendeskriptor festgehalten. Er kann vom WS-Manager generiert werden, wenn eine Nachricht von einer Anwendung in eine Warteschlange gestellt wird. Alternativ kann die Anwendung die Informationen angeben, wenn die Benutzer-ID, die der Anwendung zugeordnet ist, berechtigt ist, dies zu tun.

  Die Kontextinformationen in einer Nachricht ermöglichen es der empfangenden Anwendung, sich über den Absender der Nachricht zu informieren. Sie enthält beispielsweise den Namen der Anwendung, die die Nachricht eingibt, und die Benutzer-ID, die der Anwendung zugeordnet ist.
- Wenn ein Nachrichtenkanal gestartet wird, ist es möglich, dass der Nachrichtenkanalagent (MCA) an jedem Ende des Kanals seinen Partner authentifiziert. Dieses Verfahren wird als *gegenseitige Authentifizierung* bezeichnet. Für den sendenden Nachrichtenkanalverkehr stellt sie sicher, dass der Partner, an den Nachrichten gesendet werden sollen, authentisch ist. Für den empfangenden MCA gibt es eine ähnliche Zusicherung, dass es darum geht, Nachrichten von einem echten Partner zu empfangen.

Zugehörige Konzepte

„Identifikation und Authentifizierung” auf Seite 8
Identifikation ist die Fähigkeit, eindeutig einen Benutzer eines Systems oder einer Anwendung zu identifizieren, die im System ausgeführt wird. *Authentifizierung* ist die Möglichkeit, zu beweisen, dass ein Benutzer oder eine Anwendung wirklich die Person oder die Anwendung ist, die/der die Anwendung beansprucht.

Berechtigung in IBM MQ

Sie können Berechtigungen verwenden, um die Möglichkeiten von einzelnen Benutzern oder Anwendungen in Ihrer IBM MQ-Umgebung zu begrenzen.

Hier finden Sie einige Beispiele für die Berechtigung in einer IBM MQ-Umgebung:

- Nur ein berechtigter Administrator kann Befehle zur Verwaltung von IBM MQ-Ressourcen ausgeben.
- Eine Anwendung kann eine Verbindung zu einem WS-Manager nur herstellen, wenn die der Anwendung zugeordnete Benutzer-ID über die entsprechende Berechtigung verfügt.
- Eine Anwendung kann nur die Warteschlangen öffnen, die für ihre Funktion erforderlich sind.
- Eine Anwendung kann nur für die Themen subskribieren, die für ihre Funktion erforderlich sind.
- Die Ausführung einer Anwendung kann nur die Operationen in einer Warteschlange ausführen, die für ihre Funktion erforderlich sind. Eine Anwendung muss z. B. nur Nachrichten in einer bestimmten Warteschlange durchsuchen und keine Nachrichten einlegen oder abrufen.

Weitere Informationen zum Einrichten von Berechtigungen finden Sie im Abschnitt „Planungsberechtigung” auf Seite 93 und den zugehörigen Unterabschnitten.

Zugehörige Konzepte

„Berechtigung” auf Seite 8
Berechtigung schützt kritische Ressourcen in einem System, indem der Zugriff nur auf berechtigte Benutzer und deren Anwendungen beschränkt wird. Sie verhindert die unbefugte Verwendung einer Ressource oder die Verwendung einer Ressource in einer nicht autorisierten Weise.

Prüfung in IBM MQ

IBM MQ kann Ereignisnachrichten ausgeben, um zu erfassen, dass eine ungewöhnliche Aktivität stattgefunden hat.

Im Folgenden finden Sie einige Beispiele für die Prüfung in einer IBM MQ-Umgebung:

- Eine Anwendung versucht, eine Warteschlange zu öffnen, für die sie nicht berechtigt ist. Es wird eine Instrumentierungsereignisnachricht ausgegeben. Wenn Sie die Ereignisnachricht überprüfen, stellen Sie fest, dass dieser Versuch aufgetreten ist, und kann entscheiden, welche Aktion erforderlich ist.
- Eine Anwendung versucht, einen Kanal zu öffnen, aber der Versuch schlägt fehl, da SSL die Verbindung nicht zulässt. Es wird eine Instrumentierungsereignisnachricht ausgegeben. Wenn Sie die Ereignisnachricht überprüfen, stellen Sie fest, dass dieser Versuch aufgetreten ist, und kann entscheiden, welche Aktion erforderlich ist.

Zugehörige Konzepte

„Prüfprotokollierungs" auf Seite 9
Prüfung ist der Prozess der Aufzeichnung und Überprüfung von Ereignissen, um festzustellen, ob eine unerwartete oder unberechtigte Aktivität stattgefunden hat oder ob versucht wurde, eine solche Aktivität durchzuführen.

Vertraulichkeit in IBM MQ

Die können die Vertraulichkeit in IBM MQ durch das Verschlüsseln von Nachrichten implementieren.

Die Vertraulichkeit in einer IBM MQ-Umgebung kann folgendermaßen sichergestellt werden:

- Nachdem ein sendender Nachrichtenkanalagent eine Nachricht aus einer Übertragungswarteschlange erhalten hat, entschlüsselt IBM MQ die Nachricht mithilfe von TLS, bevor sie über das Netz an den empfangenden Nachrichtenkanalagenten gesendet wird. Am anderen Ende des Kanals wird die Nachricht entschlüsselt, bevor der empfangende MCA die Nachricht in die Zielwarteschlange einreiht.
- Solange Nachrichten in einer lokalen Warteschlange gespeichert werden, reicht das von IBM MQ bereitgestellte Verfahren zur Zugriffssteuerung aus, um die Inhalte vor nicht autorisierter Offenlegung zu schützen. Für ein höheres Maß an Sicherheit können Sie aber Advanced Message Security verwenden, um die in den Warteschlangen gespeicherten Nachrichten zu verschlüsseln.
- V 9.2.0 z/OS Nachrichten, die in lokalen Warteschlangen gespeichert sind, können im ruhenden Zustand mit der Verschlüsselung von z/OS-Datasets verschlüsselt werden.

  Weitere Informationen finden Sie im Abschnitt zur Vertraulichkeit für ruhende Daten in IBM MQ for z/OS mit der Dataset-Verschlüsselung. weitere Informationen hierzu.

Zugehörige Konzepte

„Vertraulichkeit" auf Seite 9
Der Service *Vertraulichkeit* schützt sensible Informationen vor unbefugter Offenlegung.

Datenintegrität in IBM MQ

Sie können einen Datenintegritätsservice verwenden, um festzustellen, ob eine Nachricht geändert wurde.

Die Datenintegrität kann in einer IBM MQ-Umgebung folgendermaßen sichergestellt werden:

- Sie können TLS verwenden, um festzustellen, ob der Inhalt einer Nachricht absichtlich geändert wurde, während er über ein Netz übertragen wurde. In TLS stellt der Nachrichtenauszugsalgorithmus die Erkennung geänderter Nachrichten im Transit bereit.

  Alle IBM MQ-CipherSpecs stellen einen Nachrichtenauszugsalgorithmus bereit, mit Ausnahme von TLS_RSA_WITH_NULL_NULL, der keine Integrität der Nachrichtendaten bereitstellt.

  IBM MQ erkennt geänderte Nachrichten beim Empfang. IBM MQ löst beim Empfang einer geänderten Nachricht die Fehlernachricht AMQ9661 aus und der Kanal wird gestoppt.

- Während Nachrichten in einer lokalen Warteschlange gespeichert werden, können die von IBM MQ bereitgestellten Zugriffssteuerungsmechanismen als ausreichend betrachtet werden, um eine absichtliche Änderung der Nachrichteninhalte zu verhindern.

  Für ein höheres Maß an Sicherheit können Sie jedoch Advanced Message Security verwenden, um zu ermitteln, ob die Nachrichteninhalte zwischen dem Zeitpunkt, an dem die Nachricht in die Warteschlange gestellt wurde, und dem Zeitpunkt beim Abrufen aus der Warteschlange absichtlich geändert wurden.

  Wenn eine geänderte Nachricht festgestellt wird, empfängt die Anwendung, die versucht, die Nachricht zu empfangen, einen Rückkehrcode 2063 empfängt und die Nachricht bei Verwendung eines MQGET -Aufrufs in die Warteschlange SYSTEM.PROTECTION.ERROR.QUEUE verschoben wird.

Zugehörige Konzepte

„Datenintegrität" auf Seite 9
Der *Datenintegritätsdienst* stellt fest, ob unbefugte Änderungen an Daten vorgenommen wurden.

Verschlüsselung in IBM MQ

In IBM MQ wird die Verschlüsselung mit dem TLS-Protokoll (Transport Security Layer) bereitgestellt.

Weitere Informationen finden Sie unter „TLS-Sicherheitsprotokolle in IBM MQ" auf Seite 26.

Zugehörige Konzepte

„Verschlüsselungskonzepte" auf Seite 9
In dieser Themensammlung werden die Konzepte der Verschlüsselung beschrieben, die für IBM MQ gültig sind.

TLS-Sicherheitsprotokolle in IBM MQ

IBM MQ unterstützt das TLS-Protokoll (Transport Layer Security), um die Sicherheit auf Verbindungsebene für Nachrichtenkanäle und MQI-Kanäle bereitzustellen.

Nachrichtenkanäle und MQI-Kanäle können das TLS-Protokoll verwenden, um die Sicherheit auf Verbindungsebene zu gewährleisten. Ein aufrufender MCA ist ein TLS-Client, und ein Responder MCA ist ein TLS-Server.

V 9.2.0 IBM MQ unterstützt die Versionen 1.2 und 1.3 des TLS-Protokolls. Frühere Versionen von TLS sowie SSL sind nicht standardmäßig aktiviert, können bei Bedarf aber aktiviert werden. Sie können die Verschlüsselungsalgorithmen angeben, die vom TLS-Protokoll verwendet werden, indem Sie eine CipherSpec als Teil der Kanaldefinition angeben.

V 9.2.0 Unter „CipherSpecs aktivieren" auf Seite 448 finden Sie eine Liste der CipherSpecs, die von IBM MQ und „Nicht weiter unterstützte CipherSpecs" auf Seite 465 für die veralteten CipherSpecs unterstützt werden.

Sie können die Parameter SECPROT und SSLCIPH verwenden, um das Sicherheitsprotokoll und die CipherSpec im Gebrauch auf einem Kanal anzuzeigen.

An jedem Ende eines Nachrichtenkanals und am Serverende eines MQI-Kanals agiert der MCA im Namen des Warteschlangenmanagers, mit dem er verbunden ist. Während des TLS-Handshake sendet der MCA das digitale Zertifikat des WS-Managers an seinen Partner MCA am anderen Ende des Kanals. Der IBM MQ-Code auf der Clientseite eines MQI-Kanals wird für den Benutzer der IBM MQ-Clientanwendung ausgeführt. Während des TLS-Handshakes sendet der IBM MQ-Code das digitale Zertifikat des Benutzers an den MCA auf der Serverseite des MQI-Kanals.

Warteschlangenmanagern und IBM MQ-Clientbenutzern müssen keine persönlichen digitalen Zertifikate zugeordnet sein, wenn sie als TLS-Clients ausgeführt werden, es sei denn, SSLCAUTH(REQUIRED) ist auf der Serverseite des Kanals angegeben.

Digitale Zertifikate werden in einem *Schlüsselrepository* gespeichert. Das WS-Managerattribut **SSLKeyRepository** gibt die Position des Schlüsselrepositorys an, in dem sich das digitale Zertifikat des WS-Managers befindet. Auf einem IBM MQ-Clientsystem wird mit der Umgebungsvariable MQSSLKEYR die Position des Schlüsselrepositorys angegeben, in dem sich das digitale Zertifikat des Benutzers befindet.

Alternativ kann eine IBM MQ-Clientanwendung die zugehörige Position im Feld **KeyRepository** der TLS-Konfigurationsoptionsstruktur MQSCO in einem MQCONNX-Aufruf angeben. Weitere Informationen zu Schlüsselrepositorys finden Sie in den zugehörigen Themen, und wie Sie angeben können, wo sie sich befinden.

Unterstützung für TLS

V 9.2.0 IBM MQ stellt die Unterstützung für TLS 1.2 und TLS 1.3 auf allen Plattformen bereit. Weitere Informationen zum TLS-Protokoll finden Sie in den Informationen in den Unterabschnitten.

Java- und JMS-Clients
Diese Clients verwenden die JVM, um TLS-Unterstützung bereitzustellen.

AIX, Linux, and Windows
Die TLS-Unterstützung ist mit IBM MQ installiert.

IBM i
Die TLS-Unterstützung ist ein integraler Bestandteil des IBM i-Betriebssystems.

z/OS
Die TLS-Unterstützung ist ein integraler Bestandteil des z/OS-Betriebssystems. Die TLS-Unterstützung unter z/OS wird als *System SSL* bezeichnet.

Informationen zu den Voraussetzungen für die TLS-Unterstützung in IBM MQ finden Sie unter Systemvoraussetzungen für IBM MQ.

Zugehörige Konzepte

„Verschlüsselte Sicherheitsprotokolle: TLS” auf Seite 17
Verschlüsselte Protokolle stellen sichere Verbindungen bereit, die es zwei Parteien ermöglichen, mit Datenschutz und Datenintegrität zu kommunizieren. Das TLS-Protokoll (Transport Layer Security) wurde von dem Protokoll Secure Sockets Layer (SSL) entwickelt. IBM MQ uunterstützt TLS.

Das SSL/TLS-Schlüsselrepository

Für eine gegenseitig authentifizierte TLS-Verbindung ist an jedem Ende der Verbindung ein Schlüsselrepository erforderlich. Das Schlüsselrepository enthält digitale Zertifikate und private Schlüssel.

Diese Informationen verwenden den allgemeinen Begriff *Schlüsselrepository* , um den Speicher für digitale Zertifikate und die ihnen zugeordneten privaten Schlüssel zu beschreiben. Auf das Schlüsselrepository wird von verschiedenen Namen auf verschiedenen Plattformen und Umgebungen verwiesen, die TLS unterstützen:

- IBM i Unter IBM i: *Zertifikatsspeicher*
- Unter Java und JMS: *Keystore* und *Truststore*
- ALW Unter AIX, Linux, and Windows: *Schlüsseldatenbankdatei*
- z/OS Unter z/OS: *Schlüsselring*

Weitere Informationen hierzu finden Sie unter „Digitale Zertifikate” auf Seite 12 und „Konzepte der Transport Layer Security (TLS)” auf Seite 17.

Für eine gegenseitig authentifizierte TLS-Verbindung ist an jedem Ende der Verbindung ein Schlüsselrepository erforderlich. Das Schlüsselrepository kann die folgenden Zertifikate und Anforderungen enthalten:

- Eine Reihe von CA-Zertifikaten von verschiedenen Zertifizierungsstellen, die es dem WS-Manager oder Client ermöglichen, Zertifikate zu überprüfen, die er vom Partner am fernen Ende der Verbindung empfängt. Einzelne Zertifikate können in einer Zertifikatskette enthalten sein.
- Ein oder mehrere persönliche Zertifikate, die von einer Zertifizierungsstelle empfangen wurden. Sie ordnen jedem Warteschlangenmanager oder IBM MQ MQI client ein separates persönliches Zertifikat zu. Persönliche Zertifikate sind für einen TLS-Client von wesentlicher Bedeutung, wenn die gegenseitige Authentifizierung erforderlich ist. Wenn die gegenseitige Authentifizierung nicht erforderlich ist, sind

persönliche Zertifikate auf dem Client nicht erforderlich. Das Schlüsselrepository kann auch den privaten Schlüssel enthalten, der jedem persönlichen Zertifikat entspricht.

- Zertifikatsanforderungen, die darauf warten, von einem anerkannten CA-Zertifikat signiert zu werden.

Weitere Informationen zum Schutz Ihres Schlüsselrepositorys finden Sie in „IBM MQ-Schlüsselrepositorys schützen” auf Seite 28.

Die Position des Schlüsselrepositorys hängt von der Plattform ab, die Sie verwenden:

IBM i **IBM i**

Das Schlüsselrepository ist ein Zertifikatsspeicher. Der Standardspeicher des Systemzertifikats befindet sich unter `/QIBM/UserData/ICSS/Cert/Server/Default` im Integrated File System (IFS). IBM MQ speichert das Kennwort für den Zertifikatsspeicher in einer *Kennwortstashdatei*. Die Stashdatei für den WS-Manager `QM1` ist beispielsweise `/QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth`.

Alternativ können Sie angeben, dass der IBM i-Systemzertifikatsspeicher stattdessen verwendet werden soll. Ändern Sie dazu den Wert des Attributs des Warteschlangenmanagers **SSLKEYR** in `*SYSTEM` . Dieser Wert gibt an, dass der Warteschlangenmanager den Systemzertifikatsspeicher verwenden muss, und der Warteschlangenmanager ist für die Verwendung als Anwendung mit Digital Certificate Manager (DCM) registriert.

Der Zertifikatsspeicher enthält auch den privaten Schlüssel für den WS-Manager.

ALW **AIX, Linux, and Windows-Systeme**

Das Schlüsselrepository ist eine Schlüsseldatenbankdatei. Der Name der Schlüsseldatenbankdatei muss über eine Dateierweiterung von `.kdb` verfügen. Unter AIX and Linuxlautet die Standardschlüsseldatenbankdatei für Warteschlangenmanager `QM1` beispielsweise `/var/mqm/qmgrs/QM1/ssl/key.kdb`. Wenn IBM MQ an der Standardposition installiert ist, lautet der entsprechende Pfad unter Windows `C:\ProgramData\IBM\MQ\Qmgrs\QM1\ssl\key.kdb`.

Jede Schlüsseldatenbankdatei verfügt über eine zugeordnete Kennwort-Stashdatei. Diese Datei enthält codierte Kennwörter, die es Programmen ermöglichen, auf die Schlüsseldatenbank zuzugreifen. Die Kennwortstashdatei muss sich in demselben Verzeichnis befinden, denselben Dateistamm wie die Schlüsseldatenbank haben und mit dem Suffix `.sth`enden. Beispiel: `/var/mqm/qmgrs/QM1/ssl/key.sth` .

Anmerkung: PKCS#11-Verschlüsselungshardware-Karten können die Zertifikate und Schlüssel enthalten, die ansonsten in einer Schlüsseldatenbankdatei gespeichert werden. Wenn Zertifikate und Schlüssel auf PKCS #11-Karten enthalten sind, ist für IBM MQ weiterhin Zugriff auf eine Schlüsseldatenbankdatei und eine Kennwortstashdatei erforderlich.

Auf AIX, Linux, and Windows-Systemen enthält die Schlüsseldatenbank auch den privaten Schlüssel für das persönliche Zertifikat, das dem Warteschlangenmanager oder IBM MQ MQI client zugeordnet ist.

z/OS **z/OS**

Zertifikate sind in einem Schlüsselring in z/OS enthalten.

Andere externe Sicherheitsmanager (ESMs) verwenden auch Schlüsselringe zum Speichern von Zertifikaten.

Private Schlüssel werden von RACF verwaltet.

IBM MQ-Schlüsselrepositorys schützen

Beim Schlüsselrepository für IBM MQ handelt es sich um eine Datei. Stellen Sie sicher, dass nur der vorgesehene Benutzer auf die Schlüssel-Repository-Datei zugreifen kann. Dadurch wird verhindert, dass ein Eindringling oder ein anderer nicht berechtigter Benutzer die Schlüsselrepositorydatei in ein anderes System kopiert und anschließend eine identische Benutzer-ID auf diesem System eingerichtet, um den vorgesehenen Benutzer zu imitieren.

Die Berechtigungen für die Dateien hängen von der umask des Benutzers ab und welches Tool verwendet wird. Unter Windows ist für IBM MQ-Konten die Berechtigung `BypassTraverseChecking` erforderlich, was bedeutet, dass die Berechtigungen der Ordner im Pfad keine Auswirkung haben.

Überprüfen Sie die Dateiberechtigungen der Schlüsselrepositorydateien und stellen Sie sicher, dass die Dateien und der Ordner, die den Ordner enthalten, nicht in der Welt lesbar sind, vorzugsweise nicht sogar für Gruppen lesbar.

Wenn Sie den Schlüsselspeicher schreibgeschützt machen, ist es sinnvoll, auf dem System, das Sie verwenden, nur den Administrator zu aktivieren, der Schreiboperationen aktivieren kann, um Wartungsarbeiten durchzuführen.

In der Praxis müssen Sie alle Keystores schützen, unabhängig von der Position und ob sie kennwortgeschützt sind oder nicht; schützen Sie die Schlüsselrepositorys.

Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen
Wenn Sie TLS für die Verwendung digitaler Zertifikate einrichten, müssen Sie abhängig von der verwendeten Plattform und der Methode, die Sie zum Herstellen der Verbindung verwenden, bestimmte Anforderungen für die Kennzeichnung von Kennsätze beachten.

Was ist die Zertifikatsbezeichnung?

Eine Zertifikatsbezeichnung ist eine eindeutige Kennung, die ein digitales Zertifikat darstellt, das in einem Schlüsselrepository gespeichert ist, und stellt einen geeigneten lesbaren Namen bereit, mit dem auf ein bestimmtes Zertifikat verwiesen werden kann, wenn wichtige Managementfunktionen ausgeführt werden. Sie ordnen die Zertifikatsbezeichnung zu, wenn Sie ein Zertifikat zum ersten Mal einem Schlüsselrepository hinzufügen.

Die Zertifikatsbezeichnung ist getrennt von den Feldern **`Subject Distinguished Name`** oder **`Subject Common Name`** des Zertifikats. Beachten Sie, dass **`Subject Distinguished Name`** und **`Subject Common Name`** Felder innerhalb des Zertifikats selbst sind. Diese werden definiert, wenn das Zertifikat erstellt wird und nicht geändert werden kann. Falls erforderlich, können Sie jedoch die Bezeichnung ändern, die einem digitalen Zertifikat zugeordnet ist.

Zertifikatskennsatzsyntax

Ein Zertifikatskennsatz kann Buchstaben, Zahlen und Interpunktionszeichen mit den folgenden Bedingungen enthalten:

- Multi Der Zertifikatskennsatz kann bis zu 64 Zeichen enthalten.
- z/OS Der Zertifikatskennsatz kann bis zu 32 Zeichen enthalten.
- Die Zertifikatsbezeichnung kann Leerzeichen enthalten.
- Bei Bezeichnungen muss die Groß-/Kleinschreibung beachtet
- Auf Systemen, die EBCDIC katakana verwenden, können Sie keine Kleinbuchstaben verwenden.

Zusätzliche Voraussetzungen für Zertifikatskennsatzwerte werden in den folgenden Abschnitten angegeben.

Wie wird die Zertifikatsbezeichnung verwendet?

IBM MQ verwendet Zertifikatsbezeichnungen zur Suche eines persönlichen Zertifikats, das während des TLS-Handshakes gesendet wird. Dies eliminiert Mehrdeutigkeiten, wenn mehr als ein persönliches Zertifikat im Schlüsselrepository vorhanden ist.

Sie können die Zertifikatsbezeichnung auf einen Wert Ihrer Wahl setzen. Wenn Sie keinen Wert festlegen, wird abhängig von der verwendeten Plattform ein Standardkennsatz verwendet, der auf eine Namenskonvention folgt. Weitere Informationen finden Sie in den folgenden Abschnitten zu bestimmten Plattformen.

Anmerkungen:

1. Unter Java und JMS können Sie die Zertifikatsbezeichnung nicht selbst festlegen.
2. Automatisch durch einen CHAD-Exit (Channel Automatic Definition) definierte Kanäle können die Zertifikatsbezeichnung nicht festlegen, da der TLS-Handshake bei der Kanalerstellung stattfand. Die Festlegung der Zertifikatsbezeichnung in einem CHAD-Exit für eingehende Kanäle hat keine Auswirkung.

In diesem Kontext bezieht sich ein TLS-Client auf den Verbindungspartner, der den Handshake eingeleitet hat. Dies kann ein IBM MQ-Client oder ein anderer Warteschlangenmanager sein.

Während des TLS-Handshake ruft der TLS-Client immer ein digitales Zertifikat vom Server ab und validiert es. Bei der Implementierung von IBM MQ fordert der TLS-Server immer ein Zertifikat vom Client an, und der Client stellt dem Server immer ein Zertifikat zur Verfügung, wenn ein Zertifikat gefunden wird. Wenn der Client ein persönliches Zertifikat nicht finden kann, sendet der Client eine `no certificate` -Antwort an den Server.

Der TLS-Server überprüft das Clientzertifikat immer, wenn ein Zertifikat gesendet wird. Wenn der Client kein Zertifikat sendet, schlägt die Authentifizierung fehl, wenn das Ende des Kanals, der als TLS-Server fungiert, mit dem Parameter **SSLCAUTH** definiert ist, der auf *REQUIRED* oder einen **SSLPEER**-Parametersatz gesetzt ist.

Beachten Sie, dass eingehende Kanäle (einschließlich Empfänger-, Anforderer-, Clusterempfänger-, nicht qualifizierte Server- und Serververbindungskanäle) das konfigurierte Zertifikat nur senden, wenn die IBM MQ-Version des fernen Peers die Konfiguration der Zertifikatsbezeichnung vollständig unterstützt und der Kanal ein TLS-CipherSpec verwendet.

Ein nicht qualifizierter Serverkanal ist ein Kanal, für den das Feld CONNAME nicht festgelegt wurde.

In allen anderen Fällen bestimmt der Warteschlangenmanagerparameter **CERTLABL** das gesendete Zertifikat. Insbesondere in folgenden Umgebungen wird unabhängig von der kanalspezifischen Bezeichnungseinstellung immer das durch den Parameter **CERTLABL** des Warteschlangenmanagers konfigurierte Zertifikat empfangen:

- Java und JMS-Clients unterstützen Server Name Indication (SNI), d. h. Zertifikate auf Channel-by-Channel-Basis.
- Ältere Versionen von IBM MQ als IBM MQ 8.0.
- Verwaltete .NET-Clients

Darüber hinaus muss das von einem Kanal verwendete Zertifikat für den Kanal CipherSpec geeignet sein. Weitere Informationen finden Sie im Abschnitt „Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ” auf Seite 49 .

IBM MQ 8.0 und höher unterstützt die Verwendung mehrerer Zertifikate auf demselben Queue Manager unter Verwendung einer pro-Kanal-Zertifikatsbezeichnung, die unter Verwendung des Attributs **CERTLABL** in der Kanaldefinition angegeben wird. Eingehende Kanäle zum Warteschlangenmanager (z. B. Serververbindung oder Empfänger) basieren auf der Erkennung des Kanalnamens unter Verwendung von TLS Server Name Indication (SNI), um das richtige Zertifikat vom WS-Manager zu präsentieren. Weitere Informationen zur Verwendung mehrerer Zertifikate auf einem Queue Manager finden Sie unter „So stellt IBM MQ mehrere Zertifikate zur Verfügung” auf Seite 32.

Wenn ein Kanal über IBM MQ Internet Pass-Thru (MQIPT) eine Verbindung zum Zielwarteschlangenmanager herstellt und im MQIPT-Weg sowohl **SSLServer** als auch **SSLClient** festgelegt sind, gibt es zwei separate TLS-Sitzungen zwischen den Endpunkten. In früheren Versionen als IBM MQ 9.2.5 werden die SNI-Daten nicht über die Sitzungsunterbrechung verteilt. Dadurch wird verhindert, dass ein Kanalzertifikat auf dem Zielwarteschlangenmanager für die TLS-Verbindung zwischen MQIPT und dem Queue Manager verwendet wird. V 9.2.5 Ab IBM MQ 9.2.5 kann MQIPT so konfiguriert werden, dass mehrere Zertifikate vom Zielwarteschlangenmanager verwendet werden können, indem entweder die SNI auf den Kanalnamen gesetzt wird oder indem die SNI, die an der eingehenden Verbindung zur Route empfangen wurde, übergeben wird. Weitere Informationen zur Unterstützung mehrerer Zertifikate und zu MQIPT finden Sie unter IBM MQ Unterstützung mehrerer Zertifikate mit MQIPT.

Weitere Informationen zum Verbinden eines Warteschlangenmanagers mit Einwegauthentifizierung, d. B. wenn der TLS-Client kein Zertifikat sendet, finden Sie im Abschnitt Zwei Warteschlangenmanager mit der Einwegauthentifizierung verbinden .

Multiplatforms-Systeme

Unter Multiplatforms sendet der TLS-Server ein Zertifikat an den Client.

Für WS-Manager bzw. Clients werden die folgenden Quellen in der Folge nach einem nicht leeren Wert durchsucht. Der erste nicht leere Wert bestimmt die Zertifikatsbezeichnung. Die Zertifikatsbezeichnung muss im Schlüsselrepository vorhanden sein. Wenn im richtigen Fall kein übereinstimmende Zertifikat gefunden wird und ein entsprechendes Format gefunden wird, tritt ein Fehler auf, und der TLS-Handshake schlägt fehl.

Warteschlangenmanager

1. Kennsatzattribut für Kanalzertifikat **CERTLABL**.
2. Das Kennsatzattribut des Warteschlangenmanagers **CERTLABL**.
3. Ein Standardwert, der sich im Format `ibmwebspheremq` mit dem Namen des angehängten Warteschlangenmanagers befindet, wird in Kleinbuchstaben angezeigt. Für einen WS-Manager mit dem Namen `QM1` lautet der Standardzertifikatskennsatz beispielsweise `ibmwebspheremqqm1`.

IBM MQ-Clients

1. Attribut **CERTLABL** für die Zertifikatsbezeichnung in der CLNTCONN-Kanaldefinition.
2. Attribut 'MQSCO-Struktur **`CertificateLabel`** '.
3. Umgebungsvariable **MQCERTLABL**.
4. Client- `.ini` -Datei (in ihrem SSL-Abschnitt) **`CertificateLabel`** , Attribut
5. Ein Standardwert, der im folgenden Format vorliegt: `ibmwebspheremq` mit der Benutzer-ID, die die Clientanwendung als angehängten Benutzer ausführt, alle in Kleinbuchstaben. Für eine Benutzer-ID von `USER1` lautet der Standardzertifikatskennsatz beispielsweise `ibmwebspheremquser1`.

z/OS-Systeme

IBM MQ-Clients werden unter z/OS nicht unterstützt. Ein z/OS-Warteschlangenmanager kann jedoch in der Rolle eines TLS-Clients bei der Initialisierung einer Verbindung oder eines TLS-Servers auftreten, wenn eine Verbindungsanforderung akzeptiert wird. Die Voraussetzungen für die Zertifikatsbezeichnung für z/OS-Warteschlangenmanager gelten in beiden Rollen und unterscheiden sich von den Voraussetzungen in Multiplatforms.

Für WS-Manager bzw. Clients werden die folgenden Quellen in der Folge nach einem nicht leeren Wert durchsucht. Der erste nicht leere Wert bestimmt die Zertifikatsbezeichnung. Die Zertifikatsbezeichnung muss im Schlüsselrepository vorhanden sein. Wenn im richtigen Fall kein übereinstimmende Zertifikat gefunden wird und ein entsprechendes Format gefunden wird, tritt ein Fehler auf, und der TLS-Handshake schlägt fehl.

1. Kennsatzattribut für Kanalzertifikat, **CERTLABL**.
2. Wenn sie gemeinsam genutzt wird, wird das Attribut für die Gruppe mit gemeinsamer Warteschlange **CERTQSGL** verwendet.

   Wenn keine gemeinsame Nutzung vorhanden ist, wird das Attribut "label" des Warteschlangenmanagers **CERTLABL**.
3. Ein Standardwert im Format `ibmWebSphereMQ` mit dem angehängten Namen des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange. Beachten Sie, dass diese Zeichenfolge die Groß-/Kleinschreibung beachten muss und wie gezeigt geschrieben werden muss. Für einen WS-

Manager mit dem Namen QM1 lautet der Standardzertifikatskennsatz beispielsweise ibmWebSphereMQQM1.

4. Wenn kein Zertifikat mit dem Format in Option „3" auf Seite 31 gefunden wird, versucht IBM MQ, das als Standard markierte Zertifikat im Schlüsselring zu verwenden.

Informationen zur Anzeige des Schlüsselrepositorys finden Sie unter „Schlüsselrepository für einen Warteschlangenmanager unter z/OS ermitteln" auf Seite 343.

IBM MQ Java- und IBM MQ JMS-Clients

IBM MQ Java- und IBM MQ JMS-Clients verwenden die Funktionen ihres Java Secure Socket Extension-Providers (JSSE), um während des TLS-Handshakes ein persönliches Zertifikat auszuwählen, und unterliegen daher nicht den Voraussetzungen für die Zertifikatsbezeichnung.

Das Standardverhalten ist, dass der JSSE-Client die Zertifikate im Schlüsselrepository durchläuft und das erste akzeptierbare persönliche Zertifikat ausgewählt hat. Dieses Verhalten ist jedoch nur ein Standardverhalten und hängt von der Implementierung des JSSE-Providers ab.

Darüber hinaus ist die JSSE-Schnittstelle durch Konfiguration und direkten Zugriff zur Laufzeit durch die Anwendung hochgradig anpassbar. Einzelheiten finden Sie in der Dokumentation, die Ihr JSSE-Provider zur Verfügung gestellt hat.

Zur Fehlerbehebung, bzw. wenn Sie das von der IBM MQ Java-Clientanwendung in Verbindung mit dem JSSE-Provider durchgeführte Handshaking besser verstehen möchten, können Sie mit `javax.net.debug=ssl` das Debugging in der JVM-Umgebung aktivieren.

Sie können die Variable in der Anwendung, durch Konfiguration oder durch Eingabe von `-Djavax.net.debug=ssl` in der Befehlszeile festlegen.

Linux *So stellt IBM MQ mehrere Zertifikate zur Verfügung*

Die Servernamensanzeige (Server Name Indication, SNI) ist eine Erweiterung des TLS-Protokolls, die es einem Client ermöglicht, anzugeben, welchen Service er benötigt. In der IBM MQ-Terminologie ist dies einem Kanal gleichzusetzen.

Die SNI-Erweiterung wird von IBM MQ verwendet, um zu ermöglichen, dass mehrere Zertifikate über verschiedene Kanäle mit dem Parameter CERTLABL in der Kanaldefinition angegeben werden können.

Die von IBM MQ verwendete SNI-Adresse basiert auf dem Kanalnamen, der angefordert wird, gefolgt von einem Suffix von `.chl.mq.ibm.com`.

Die Namen von IBM MQ-Kanalnamen werden als gültige SNI-Namen wie folgt zugeordnet:

- Großbuchstaben von A bis Z werden in Kleinbuchstaben umgesetzt
- Die Ziffern 0 bis 9 bleiben unverändert
- Alle anderen Zeichen, einschließlich der Kleinbuchstaben a bis z, werden in ihren zweistelligen hexadezimalen ASCII-Zeichencode (in Kleinbuchstaben) konvertiert, gefolgt von einem Bindestrich.
  - Kleinbuchstaben von a bis z werden hexadezimal `61-` bzw. `7a-` zugeordnet
  - Prozent (%) wird hexadezimal `25-` zugeordnet
  - Bindestrich (-) wird hexadezimal `2d-` zugeordnet
  - Punkt (.) wird hexadezimal `2e-` zugeordnet
  - Schrägstrich (/) wird hexadezimal `2f-` zugeordnet
  - Unterstrich (_) wird hexadezimal `5f-` zugeordnet

Auf EBCDIC-Plattformen wird der Kanalname in ASCII konvertiert, bevor diese Zuordnung angewendet wird.

Als Beispiel wird der Kanalname `TO.QMGR1` einer SNI-Adresse von `to2e-qmgr1.chl.mq.ibm.com`zugeordnet.

Im Gegensatz dazu ordnet der Kanalname `to.qmgr1` in Kleinbuchstaben die SNI-Adresse `74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com` zu.

Anmerkung: In Umgebungen, in denen die generierte SNI-URL den URL-Formatierungsspezifikationen entsprechen muss, z. B. wenn ein Client eine Verbindung zu einem Queue Manager herstellt, der in Red Hat® OpenShift® über eine Red Hat OpenShift-Route ausgeführt wird, darf der Kanalname nicht mit einem Kleinbuchstaben enden.

Mit der zusätzlichen Eigenschaft **OutboundSNI** der SSL-Zeilengruppe können Sie auswählen, ob die SNI auf den Namen des IBM MQ-Zielkanals für das ferne System gesetzt werden soll, wenn eine TLS-Verbindung eingeleitet wird, oder auf den Hostnamen. Weitere Informationen zur Eigenschaft **OutboundSNI** finden Sie unter SSL-Zeilengruppe der Datei 'qm.ini' und SSL-Zeilengruppe der Clientkonfigurationsdatei.

Für mehrere Zertifikate ist es erforderlich, dass die SNI auf den IBM MQ -Kanalnamen gesetzt ist. Wenn ein Hostname, eine angepasste oder keine SNI verwendet wird, um eine Verbindung zu einem IBM MQ -Kanal mit einer konfigurierten Zertifikatsbezeichnung herzustellen, wird die verbindende Anwendung mit MQRC_SSL_INITIALIZATION_ERROR abgelehnt und eine Nachricht AMQ9673 wird in den Fehlerprotokollen des fernen Warteschlangenmanagers ausgegeben.

V 9.2.5 Wenn ein Kanal über IBM MQ Internet Pass-Thru (MQIPT) eine Verbindung zum Zielwarteschlangenmanager herstellt, muss MQIPT so konfiguriert sein, dass entweder die SNI auf den Kanalnamen gesetzt oder die SNI, die an der eingehenden Verbindung empfangen wurde, an die Route übergeben wird, damit mehrere Zertifikate vom Zielwarteschlangenmanager verwendet werden können. Weitere Informationen zur Unterstützung mehrerer Zertifikate und zu MQIPT finden Sie unter IBM MQ-Unterstützung für mehrere Zertifikate mit MQIPT.

Weitere Informationen zur Verwendung dieser Eigenschaft finden Sie im Abschnitt Verbindung zu einem Warteschlangenmanager herstellen, der in einem Red Hat OpenShift-Cluster implementiert ist.

Das Schlüsselrepository des Warteschlangenmanagers wird neu freigegeben.
Wenn Sie den Inhalt eines Schlüsselrepositorys ändern, wird der neue Inhalt vom WS-Manager nicht sofort ausgewählt. Damit ein WS-Manager den Inhalt des neuen Schlüsselrepositorys verwenden kann, müssen Sie den Befehl REFRESH SECURITY TYPE (SSL) absetzen.

Dieser Prozess ist beabsichtigt und verhindert die Situation, in der mehrere aktive Kanäle unterschiedliche Versionen eines Schlüsselrepositorys verwenden können. Als Sicherheitssteuerung kann nur eine Version eines Schlüsselrepositorys vom WS-Manager geladen werden.

Weitere Informationen zum Befehl REFRESH SECURITY TYPE (SSL) finden Sie in REFRESH SECURITY .

Sie können ein Schlüsselrepository auch mit PCF-Befehlen oder dem IBM MQ Explorer aktualisieren. Weitere Informationen finden Sie unter Befehl MQCMD_REFRESH_SECURITY und im Abschnitt *TLS-Sicherheit aktualisieren* zum IBM MQ Explorer dieser Produktdokumentation.

Zugehörige Konzepte
„Clientansicht des SSL/TLS-Schlüsselrepositoryinhalts und der SSl/TLS-Einstellungen neu anzeigen” auf Seite 33
Wenn Sie die Clientanwendung mit dem aktualisierten Inhalt des Schlüsselrepositorys aktualisieren möchten, müssen Sie die Clientanwendung stoppen und erneut starten.

Clientansicht des SSL/TLS-Schlüsselrepositoryinhalts und der SSl/TLS-Einstellungen neu anzeigen
Wenn Sie die Clientanwendung mit dem aktualisierten Inhalt des Schlüsselrepositorys aktualisieren möchten, müssen Sie die Clientanwendung stoppen und erneut starten.

Sie können die Sicherheit auf einem IBM MQ-Client nicht aktualisieren. Es gibt keine Entsprechung des Befehls REFRESH SECURITY TYPE(SSL) für Clients (weitere Informationen finden Sie unter REFRESH SECURITY).

Sie müssen die Anwendung stoppen und erneut starten, wenn Sie das Sicherheitszertifikat ändern, um die Clientanwendung mit dem aktualisierten Inhalt des Schlüsselrepositorys zu aktualisieren.

Wenn der Kanal erneut gestartet wird und die Konfigurationen aktualisiert werden, ist es möglich, dass Sie die Sicherheit auf dem Client aktualisieren können, indem Sie den Befehl STOP CHL STATUS (INACTIVE) ausgeben.

Zugehörige Konzepte

„Das Schlüsselrepository des Warteschlangenmanagers wird neu freigegeben." auf Seite 33
Wenn Sie den Inhalt eines Schlüsselrepositorys ändern, wird der neue Inhalt vom WS-Manager nicht sofort ausgewählt. Damit ein WS-Manager den Inhalt des neuen Schlüsselrepositorys verwenden kann, müssen Sie den Befehl REFRESH SECURITY TYPE (SSL) absetzen.

MQCSP-Kennwortschutz

Ab IBM MQ 8.0 können Sie Kennwörter, die in der MQCSP-Struktur enthalten sind, mit der IBM MQ-Funktion geschützt oder durch die TLS-Verschlüsselung verschlüsselt senden.

Wichtig: Der MQCSP-Kennwortschutz ist für Test-und Entwicklungszwecke nützlich, da die Verwendung des MQCSP-Kennwortschutzes einfacher ist, als die TLS-Verschlüsselung zu konfigurieren, aber nicht als sicher. Zu Produktionszwecken sollten Sie die TLS-Verschlüsselung dem Kennwortschutz mit IBM MQ vorziehen, insbesondere dann, wenn das Netz zwischen dem Client und dem Warteschlangenmanager nicht vertrauenswürdig ist, da die TLS-Verschlüsselung sicherer ist.

Wenn Sie genau darüber besorgt sind, welche Verschlüsselung verwendet wird und wie viel Schutz sie bietet, müssen Sie die vollständige TLS-Verschlüsselung verwenden. In dieser Situation sind die Algorithmen öffentlich bekannt, und Sie können die geeignete Algorithmen für Ihr Unternehmen auswählen, indem Sie das Kanalattribut **SSLCIPH** verwenden.

Weitere Informationen zur MQCSP-Struktur finden Sie in der MQCSP-Struktur .

Der Kennwortschutz wird verwendet, wenn alle folgenden Bedingungen erfüllt sind:

- Beide Enden der Verbindung verwenden IBM MQ 8.0 oder höher.
- Der Kanal verwendet die TLS-Verschlüsselung nicht. Ein Kanal verwendet keine TLS-Verschlüsselung, wenn der Kanal ein leeres Attribut **SSLCIPH** hat, oder das Attribut **SSLCIPH** auf eine CipherSpec gesetzt ist, die keine Verschlüsselung bereitstellt. Null-Chiffrierwerte, z. B. NULL_SHA, stellen keine Verschlüsselung bereit.
- Sie haben **MQCSP**festgelegt.**AuthenticationType** in MQCSP_AUTH_USER_ID_AND_PWD. Wenn Sie diesen Wert festlegen, können weitere Prüfungen ausgewertet werden, um zu entscheiden, ob der Kennwortschutz ausgeführt wird. Der Standardwert **MQCSP**.**AuthenticationType** ist MQCSP_AUTH_NONE. Bei der Standardeinstellung besteht kein Kennwortschutz. Weitere Informationen finden Sie unter **AuthenticationType**.
- Wenn es sich beim Client um den IBM MQ Explorer handelt und der Kompatibilitätsmodus für die Benutzeridentifikation nicht aktiviert ist, was die Standardeinstellung ist. Diese Bedingung gilt nur für den IBM MQ Explorer.

Wenn diese Bedingungen nicht erfüllt sind, wird das Kennwort in Klartext gesendet, sofern dies nicht durch die Konfigurationseinstellung von **PasswordProtection** untersagt ist.

Konfigurationseinstellung für PasswordProtection

Das Attribut **PasswordProtection** im Abschnitt "Kanäle" der Konfigurationsdateien des Clients und des Warteschlangenmanagers .ini kann verhindern, dass Kennwörter in Klartext gesendet werden. Das Attribut kann einen der folgenden Werte annehmen. Der Standardwert ist `compatible`:

kompatibel
Das Kennwort kann in Klartext gesendet werden, wenn der Warteschlangenmanager oder der Client auf einer früheren Version von IBM MQ als IBM MQ 8.0 ausgeführt werden. Dies bedeutet, dass Klartextkennwörter aus Gründen der Kompatibilität zulässig sind.

Daher gilt Folgendes:

- Das Kennwort wird von der TLS-CipherSpec verschlüsselt gesendet, wenn die TLS-Verschlüsselung verwendet wird und die CipherSpec nicht null ist.

- Das Kennwort kann in Klartext gesendet werden, wenn der Warteschlangenmanager oder der Client auf einer früheren Version von IBM MQ als IBM MQ 8.0 ausgeführt werden und die TLS-Verschlüsselung nicht verwendet wird. Das Kennwort wird in Klartext gesendet, da die Versionen von IBM MQ vor IBM MQ 8.0 Kennwörter nur in Klartext senden können.
- Das Kennwort wird geschützt gesendet, wenn sowohl der Warteschlangenmanager als auch der Client eine Version von IBM MQ at IBM MQ 8.0 oder höher ausführen und entweder eine CipherSpec mit Nullwert verwendet wird oder keine TLS-Verschlüsselung verwendet wird. In: **MQCSP**.**AuthenticationType** muss auf MQCSP_AUTH_USER_ID_AND_PWD gesetzt werden.
- Die Verbindung schlägt fehl, bevor das Kennwort gesendet wird, wenn sowohl der Warteschlangenmanager als auch der Client eine Version von IBM MQ mit IBM MQ 8.0 oder höher und **MQCSP**ausführen.**AuthenticationType** ist nicht auf MQCSP-AUTH_USER_ID_AND_PWD gesetzt.

immer
Das Kennwort muss entweder mit einer CipherSpec verschlüsselt werden, die keine Null- CipherSpecist, oder **MQCSP**.**AuthenticationType** muss auf MQCSP_AUTH_USER_ID_AND_PWD gesetzt werden. Andernfalls schlägt die Verbindung fehl. Dies bedeutet, dass Klartextkennwörter nicht zulässig sind.

Daher gilt Folgendes:

- Das Kennwort wird von der TLS-CipherSpec verschlüsselt gesendet, wenn die TLS-Verschlüsselung verwendet wird und die CipherSpec nicht null ist.
- Das Kennwort wird geschützt gesendet, wenn sowohl der Warteschlangenmanager als auch der Client eine Version von IBM MQ at IBM MQ 8.0 oder höher ausführen und entweder die TLS-Verschlüsselung nicht verwendet wird oder eine Null- CipherSpec verwendet wird. In: **MQCSP**.**AuthenticationType** muss auf MQCSP_AUTH_USER_ID_AND_PWD gesetzt werden.
- Die Verbindung schlägt fehl, bevor das Kennwort gesendet wird, wenn der Warteschlangenmanager oder der Client auf einer Version von IBM MQ vor IBM MQ 8.0 ausgeführt werden und die TLS-Verschlüsselung nicht verwendet wird. Die Verbindung schlägt fehl, da die Versionen von IBM MQ vor IBM MQ 8.0 Kennwörter nur in Klartext senden können und das Kennwort für `always` verschlüsselt oder geschützt sein muss.

optional
Das Kennwort kann optional geschützt gesendet werden, wird aber in Klartext gesendet, wenn **MQCSP**angegeben ist.**AuthenticationType** ist nicht auf MQCSP-AUTH_USER_ID_AND_PWD gesetzt. Dies bedeutet, dass Klartextkennwörter von einem beliebigen Client gesendet werden dürfen.

Daher gilt Folgendes:

- Das Kennwort wird von der TLS-CipherSpec verschlüsselt gesendet, wenn die TLS-Verschlüsselung verwendet wird und die CipherSpec nicht null ist.
- Das Kennwort wird in Klartext gesendet, wenn eine CipherSpec mit Nullwert und **MQCSP**verwendet wird.**AuthenticationType** ist nicht auf MQCSP-AUTH_USER_ID_AND_PWD gesetzt.
- Das Kennwort kann in Klartext gesendet werden, wenn der Warteschlangenmanager oder der Client auf einer früheren Version von IBM MQ als IBM MQ 8.0 ausgeführt werden und die TLS-Verschlüsselung nicht verwendet wird. Das Kennwort wird in Klartext gesendet, da die Versionen von IBM MQ vor IBM MQ 8.0 Kennwörter nur in Klartext senden können.
- Das Kennwort wird geschützt gesendet, wenn sowohl der Warteschlangenmanager als auch der Client eine Version von IBM MQ at IBM MQ 8.0 oder höher ausführen, keine TLS-Verschlüsselung verwendet wird oder eine Null- CipherSpec verwendet wird und **MQCSP**.**AuthenticationType** wird auf MQCSP_AUTH_USER_ID_AND_Partnerkennwort gesetzt.

Warnung
Klartextkennwörter dürfen von einem beliebigen Client gesendet werden. Wenn ein Klartextkennwort empfangen wird, wird eine Warnung (AMQ9297) in die Fehlerprotokolle des Warteschlangenmanagers geschrieben.

Für Java-und JMS-Clients ändert sich das Verhalten der **PasswordProtection** -Attributänderungen abhängig von der Wahl des Kompatibilitätsmodus oder des MQCSP-Modus:

- Wenn Java-und JMS-Clients im Kompatibilitätsmodus betrieben werden, wird während der Verbindungsverarbeitung keine MQCSP-Struktur mit einem Flowing-Fehler ausgeführt. Daher entspricht das Verhalten des Attributs **PasswordProtection** dem Verhalten, das für Clients beschrieben wurde, die auf einer Version von IBM MQ vor IBM MQ 8.0 ausgeführt werden.
- Wenn Java-und JMS-Clients im MQCSP-Modus betrieben werden, ist das Verhalten des Attributs **PasswordProtection** das Verhalten wie beschrieben.

Weitere Informationen zur Verbindungsauthentifizierung mit Java und JMS-Clients finden Sie unter „Verbindungsauthentifizierung mit dem Java-Client" auf Seite 86.

Digital Certificate Manager (DCM)

Mit dem DCM können Sie digitale Zertifikate und private Schlüssel unter IBM i verwalten.

Mit dem Digital Certificate Manager (DCM) können Sie digitale Zertifikate verwalten und diese in gesicherten Anwendungen auf dem IBM i-Server verwenden. Mit Digital Certificate Manager können Sie digitale Zertifikate von Zertifizierungsstellen (CAs) oder anderen Drittanbietern anfordern und verarbeiten. Sie können auch als lokale Zertifizierungsinstanz fungieren, um digitale Zertifikate für Ihre Benutzer zu erstellen und zu verwalten.

DCM unterstützt auch die Verwendung von Zertifikatswiderrufslisten (Certificate Revocation Lists, CRLs), um ein strenges Zertifikats-und Anwendungsvalidierungsprozess bereitzustellen. Mit dem DCM können Sie die Position definieren, an der sich eine bestimmte CRL der Zertifizierungsstelle auf einem LDAP-Server befindet, damit IBM MQ bestätigen kann, dass ein bestimmtes Zertifikat nicht widerrufen wurde.

DCM unterstützt und kann Zertifikate in einer Vielzahl von Formaten automatisch erkennen. Wenn DCM ein PKCS#12-codiertes Zertifikat oder ein PKCS#7-Zertifikat erkennt, das verschlüsselte Daten enthält, fordert es den Benutzer automatisch auf, das Kennwort einzugeben, das zum Verschlüsseln des Zertifikats verwendet wurde. DCM fordert keine PKCS#7-Zertifikate an, die keine verschlüsselten Daten enthalten.

DCM stellt eine browserbasierte Benutzerschnittstelle bereit, mit der Sie digitale Zertifikate für Ihre Anwendungen und Benutzer verwalten können. Die Benutzerschnittstelle ist in zwei Hauptrahmen unterteilt: ein Navigationsrahmen und ein Taskrahmen.

Sie verwenden den Navigationsrahmen, um die Tasks zum Verwalten von Zertifikaten oder Anwendungen auszuwählen, die sie verwenden. Einige einzelne Tasks werden direkt im Hauptnavigationsrahmen angezeigt, die meisten Tasks im Navigationsrahmen sind jedoch in Kategorien unterteilt. Beispiel: "Zertifikate verwalten" ist eine Taskkategorie, die verschiedene einzelne geführte Tasks enthält, z. B. "Zertifikat anzeigen", "Zertifikat erneuern" und "Zertifikat importieren". Wenn ein Element im Navigationsrahmen eine Kategorie ist, die mehr als eine Aufgabe enthält, wird links davon ein Pfeil angezeigt. Der Pfeil zeigt an, dass beim Auswählen des Kategorielinks eine erweiterte Liste mit Tasks angezeigt wird, in der Sie die auszuführende Task auswählen können.

Wichtige Informationen zu DCM finden Sie in den folgenden Veröffentlichungen zu IBM Redbooks:

- *IBM i Wired Network Security: OS/400 V5R1 DCM and Cryptographic Enhancements*, SG24-6168. Beachten Sie insbesondere die Anhänge, die wichtige Informationen zur Einrichtung Ihres IBM i-Systems als lokale Zertifizierungsstelle enthalten.
- *AS/400 Internet Security: Developing a Digital Certificate Infrastructure* , IBM Form SG24-5659. Für genauere Informationen, siehe Kapitel 5. *Digital Certificate Manager für AS/400*, in dem das AS/400-DCM erläutert wird.

Federal Information Processing Standards (FIPS)

In diesem Abschnitt wird das FIPS-Verschlüsselungsprogramm (FIPS Cryptomodule Validation Program) des National Institute of Standards and Technology (US National Institute of Standards and Technology) und die Verschlüsselungsfunktionen eingeführt, die auf TLS-Kanälen verwendet werden können.

Anmerkung: Unter AIX, Linux, and Windowsstellt IBM MQ Konformität mit FIPS 140-2 über das Verschlüsselungsmodul "IBM Crypto for C" bereit. Das Zertifikat für dieses Modul wurde in den Langzeitstatus versetzt. Kunden sollten das IBM Crypto for C-Zertifikat anzeigen und sich über Empfehlungen

von NIST im Klaren sein. Ein Ersatz-FIPS 140-3-Modul ist derzeit in Bearbeitung und sein Status kann angezeigt werden, indem in der NIST-CMVP-Module in der Prozesslistenach ihm gesucht wird.

Diese Informationen gelten für die folgenden Plattformen:

- ALW AIX, Linux, and Windows
- z/OS z/OS

ALW Weitere Informationen zur FIPS 140-2-Konformität einer TLS-Verbindung von IBM MQ unter AIX, Linux, and Windows finden Sie unter „Federal Information Processing Standards (FIPS) für AIX, Linux, and Windows" auf Seite 37.

z/OS Weitere Informationen zur FIPS 140-2-Konformität einer TLS-Verbindung von IBM MQ unter z/OS finden Sie unter „Federal Information Processing Standards (FIPS) für z/OS" auf Seite 40.

Wenn Verschlüsselungshardware vorhanden ist, werden von IBM MQ die vom Hardwarehersteller bereitgestellten Verschlüsselungsmodul verwendet. Ist dies der Fall, ist die Konfiguration nur FIPS-konform, wenn die Verschlüsselungsmodule FIPS-zertifiziert sind.

Im Laufe der Zeit werden die Federal Information Processing Standards aktualisiert, um neue Angriffe auf Verschlüsselungsalgorithmen und -protokolle zu widerzuspiegeln. Einige CipherSpecs können zum Beispiel nicht mehr FIPS-zertifiziert sein. Wenn solche Änderungen auftreten, wird IBM MQ ebenfalls aktualisiert, um den neuesten Standard zu implementieren. Daher werden nach der Anwendung der Wartung möglicherweise Änderungen im Verhalten angezeigt.

Zugehörige Konzepte

„Angeben, dass nur FIPS-zertifizierte CipherSpecs während der Ausführung auf dem MQI-Client verwendet werden" auf Seite 290
Erstellen Sie Ihre Schlüsselrepositorys mit FIPS-konformer Software und geben Sie dann an, dass der Kanal FIPS-zertifizierte CipherSpecs verwenden muss.

„runmqckm, runmqakmund strmqikm für die Verwaltung digitaler Zertifikate verwenden" auf Seite 307
Verwalten Sie Schlüssel und digitale Zertifikate auf Systemen mit AIX, Linux, and Windows über die GUI **strmqikm** (iKeyman) oder über die Befehlszeile mit **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd).

Zugehörige Tasks

TLS in IBM MQ classes for Java aktivieren

Transport Layer Security (TLS) mit IBM MQ classes for JMS verwenden

Zugehörige Verweise

TLS-Eigenschaften von JMS-Objekten

„Federal Information Processing Standards" auf Seite 22
Die US-Regierung produziert technische Beratung zu IT-Systemen und Sicherheit, einschließlich der Datenverschlüsselung. Das National Institute for Standards and Technology (NIST) ist ein wichtiges Gremium, das sich mit IT-Systemen und der Sicherheit befasst. NIST erstellt Empfehlungen und Standards, einschließlich der Federal Information Processing Standards (FIPS).

ALW *Federal Information Processing Standards (FIPS) für AIX, Linux, and Windows*

Wenn die Verschlüsselung in einem SSL/TLS-Kanal auf AIX, Linux, and Windows-Systemen erforderlich ist, verwendet IBM MQ ein Verschlüsselungspaket mit dem Namen „IBM Crypto for C (ICC)". Auf AIX, Linux, and Windows-Plattformen erfüllt die ICC-Software das Federal Information Processing Standards (FIPS) Cryptomodule Validation Program des US National Institute of Standards and Technology (Federal Information Processing Standards) auf Ebene 140-2.

Anmerkung: Unter AIX, Linux, and Windowsstellt IBM MQ Konformität mit FIPS 140-2 über das Verschlüsselungsmodul "IBM Crypto for C" bereit. Das Zertifikat für dieses Modul wurde in den Langzeitstatus versetzt. Kunden sollten das IBM Crypto for C-Zertifikat anzeigen und sich über Empfehlungen von NIST im Klaren sein. Ein Ersatz-FIPS 140-3-Modul ist derzeit in Bearbeitung und sein Status kann angezeigt werden, indem in der NIST-CMVP-Module in der Prozesslistenach ihm gesucht wird.

Die FIPS-140-2-Konformität einer IBM MQ-TLS-Verbindung auf AIX, Linux, and Windows-Systemen lautet wie folgt:

- Für alle IBM MQ-Nachrichtenkanäle (außer CLNTCONN-Kanaltypen) ist die Verbindung FIPS-konform, wenn die folgenden Bedingungen erfüllt sind:
  - Die installierte GSKit-ICC-Version ist FIPS 140-2-konform mit der installierten Version und Hardwarearchitektur des Betriebssystems.
  - Das Attribut SSLFIPS des WS-Managers wurde auf YES gesetzt.
  - Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .
- Für alle IBM MQ MQI client-Anwendungen verwendet die Verbindung GSKit und ist FIPS-konform, wenn die folgenden Bedingungen erfüllt sind:
  - Die installierte GSKit-ICC-Version ist FIPS 140-2-konform mit der installierten Version und Hardwarearchitektur des Betriebssystems.
  - Sie haben angegeben, dass nur FIPS-zertifizierte Verschlüsselung verwendet werden soll, wie in dem zugehörigen Thema für den MQI-Client beschrieben.
  - Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .
- Für IBM MQ classes for Java-Anwendungen, die den Clientmodus verwenden, nutzt die Verbindung die TLS-Implementierungen der JRE und ist FIPS-konform, wenn die folgenden Bedingungen erfüllt sind:
  - Die Java Runtime Environment, mit der die Anwendung ausgeführt wird, ist FIPS-konform mit der installierten Betriebssystemversion und der Hardwarearchitektur.
  - Sie haben angegeben, dass nur FIPS-zertifizierte Verschlüsselung verwendet werden soll, wie in dem zugehörigen Abschnitt zum Java-Client beschrieben wird.
  - Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .
- Für IBM MQ classes for JMS-Anwendungen, die den Clientmodus verwenden, nutzt die Verbindung die TLS-Implementierungen der JRE und ist FIPS-konform, wenn die folgenden Bedingungen erfüllt sind:
  - Die Java Runtime Environment, mit der die Anwendung ausgeführt wird, ist FIPS-konform mit der installierten Betriebssystemversion und der Hardwarearchitektur.
  - Sie haben angegeben, dass nur FIPS-zertifizierte Verschlüsselung verwendet werden soll, wie in dem zugehörigen Abschnitt zum JMS-Client beschrieben wird.
  - Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .
- Bei nicht verwalteten .NET-Clientanwendungen verwendet die Verbindung GSKit und ist FIPS-konform, wenn die folgenden Bedingungen erfüllt sind:
  - Die installierte GSKit-ICC-Version ist FIPS 140-2-konform mit der installierten Version und Hardwarearchitektur des Betriebssystems.
  - Sie haben angegeben, dass nur FIPS-zertifizierte Verschlüsselung verwendet werden soll, wie in dem zugehörigen Abschnitt zum .NET-Client beschrieben wird.
  - Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .
- Bei nicht verwalteten .NET-Clientanwendungen von XMS verwendet die Verbindung GSKit und ist FIPS-konform, wenn die folgenden Bedingungen erfüllt sind:
  - Die installierte GSKit-ICC-Version ist FIPS 140-2-konform mit der installierten Version und Hardwarearchitektur des Betriebssystems.
  - Sie haben angegeben, dass nur FIPS-zertifizierte Verschlüsselung verwendet werden soll, wie in der Dokumentation zu XMS .NET beschrieben.

– Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .

Alle unterstützten Plattformen sind FIPS 140-2-zertifiziert, mit Ausnahme der in der Readme-Datei enthaltenen Readme-Datei, die in den einzelnen Fixpacks oder Refresh-Packs enthalten ist.

Bei TLS-Verbindungen mit GSKit wird die Komponente, die FIPS 140-2-zertifiziert ist, als *ICC* bezeichnet. Es handelt sich um die Version dieser Komponente, die die Konformität von GSKit FIPS auf einer bestimmten Plattform bestimmt. Führen Sie den Befehl **dspmqver -p 64 -v** aus, um die derzeit installierte ICC-Version zu ermitteln.

Das folgende Beispiel zeigt einen Auszug der **dspmqver -p 64 -v** -Ausgabe, die sich auf ICC bezieht:

```
ICC
============
@(#)CompanyName:   IBM Corporation
@(#)LegalTrademarks: IBM
@(#)Dateibeschreibung: IBM Crypto für Programmiersprache C
@(#)FileVersion:   8.0.0.0
@ (#) LegalCopyright: Lizenziertes Material-Eigentum von IBM
@ (#) ICC
@ (#) (C) Copyright IBM Corp. 2002, 2024.
@ (#) Alle Rechte vorbehalten. Benutzer der US-Regierung
@ (#) Restricted Rights-Use, duplication or disclosure
@(#)         restricted by GSA ADP Schedule Contract with IBM Corp.
@ (#) Produktname: icc_8.0 (GoldCoast Build) 100415
@(#)ProductVersion:  8.0.0.0
@(#)ProductInfo:   10/04/15.03:32:19.10/04/15.18:41:51
@ (#) CMVCInfo:
```

Die NIST-Zertifizierungsanweisung für GSKit ICC 8 (in GSKit 8 enthalten) finden Sie unter der folgenden Adresse: Cryptographic Module Validation Program.

Wenn Verschlüsselungshardware vorhanden ist, werden von IBM MQ die vom Hardwarehersteller bereitgestellten Verschlüsselungsmodul verwendet. Ist dies der Fall, ist die Konfiguration nur FIPS-konform, wenn die Verschlüsselungsmodule FIPS-zertifiziert sind.

Bei Einhaltung der FIPS 140-2-Konformität erzwungene Triple DES-Einschränkungen

Wenn IBM MQ für die Einhaltung von FIPS 140-2 konfiguriert ist, werden zusätzliche Einschränkungen in Bezug auf Triple DES (3DES) CipherSpecs umgesetzt. Diese Einschränkungen ermöglichen die Einhaltung der Empfehlung NIST SP800-67 der USA.

1. Alle Teile des Triple DES-Schlüssels müssen eindeutig sein.
2. Kein Teil des Triple DES-Schlüssels kann ein Weak-, Semi-Weak-oder Possibly-Weak-Schlüssel sein, entsprechend den Definitionen in NIST SP800-67.
3. Es können nicht mehr als 32 GB Daten über die Verbindung übertragen werden, bevor ein geheimer Schlüssel zurückgesetzt werden muss. Standardmäßig setzt IBM MQ den geheimen Sitzungsschlüssel nicht zurück, so dass dieses Zurücksetzen konfiguriert werden muss. Wenn die Verwendung einer Triple DES-CipherSpec-und FIPS 140-2-Konformitätserfolgung nicht aktiviert wird, wird die Verbindung mit dem Fehler AMQ9288 nach der Überschreitung der maximalen Bytezahl mit dem Fehler AMQ9288 geschlossen. Informationen zum Konfigurieren der Zurücksetzung von geheimen Schlüsseln finden Sie im Abschnitt „Zurücksetzen von geheimen SSL-und TLS-Schlüsseln” auf Seite 497.

IBM MQ generiert Triple DES-Sitzungsschlüssel, die bereits den Regeln 1 und 2 entsprechen. Um die dritte Einschränkung zu erfüllen, müssen Sie jedoch die Zurücksetzung des geheimen Schlüssels aktivieren, wenn Triple DES CipherSpecs in einer FIPS 140-2-Konfiguration verwendet wird. Alternativ können Sie Triple DES nicht verwenden.

Zugehörige Konzepte

„Angeben, dass nur FIPS-zertifizierte CipherSpecs während der Ausführung auf dem MQI-Client verwendet werden” auf Seite 290
Erstellen Sie Ihre Schlüsselrepositorys mit FIPS-konformer Software und geben Sie dann an, dass der Kanal FIPS-zertifizierte CipherSpecs verwenden muss.

„runmqckm, runmqakmund strmqikm für die Verwaltung digitaler Zertifikate verwenden” auf Seite 307
Verwalten Sie Schlüssel und digitale Zertifikate auf Systemen mit AIX, Linux, and Windows über die GUI **strmqikm** (iKeyman) oder über die Befehlszeile mit **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd).

Zugehörige Tasks

TLS in IBM MQ classes for Java aktivieren

Transport Layer Security (TLS) mit IBM MQ classes for JMS verwenden

Zugehörige Verweise

TLS-Eigenschaften von JMS-Objekten

„Federal Information Processing Standards” auf Seite 22
Die US-Regierung produziert technische Beratung zu IT-Systemen und Sicherheit, einschließlich der Datenverschlüsselung. Das National Institute for Standards and Technology (NIST) ist ein wichtiges Gremium, das sich mit IT-Systemen und der Sicherheit befasst. NIST erstellt Empfehlungen und Standards, einschließlich der Federal Information Processing Standards (FIPS).

z/OS *Federal Information Processing Standards (FIPS) für z/OS*

Wenn die Verschlüsselung in einem SSL/TLS-Kanal unter z/OS erforderlich ist, verwendet IBM MQ einen Service, der als System SSL bezeichnet wird. Das Ziel von System SSL ist es, die Fähigkeit zur sicheren Ausführung in einem Modus bereitzustellen, der so konzipiert ist, dass er das FIPS-Validierungsprogramm (Federal Information Processing Standards) des National Institute of Standards and Technology (FIPS) des US-amerikanischen National Institute of Standards and Technology auf der Stufe 140-2 erfüllen kann.

Bei der Implementierung von FIPS 140-2-konformen Verbindungen mit IBM MQ-TLS-Verbindungen müssen Sie eine Reihe von Punkten berücksichtigen:

- Stellen Sie sicher, dass die folgenden Bedingungen erfüllt sind, um IBM MQ-Nachrichtenkanäle für die FIPS-Konformität zu aktivieren:
  - System SSL Security Level 3 FMID wird installiert und konfiguriert (siehe Installation von IBM MQ planen).
  - System-SSL-Module werden validiert.
  - Das Attribut "SSLFIPS" des WS-Managers wurde auf **YES** gesetzt.

Bei der Ausführung im FIPS-Modus nutzt System SSL den CP-Assistenten für Verschlüsselungsfunktion (CPACF), wenn verfügbar. Verschlüsselungsfunktionen, die von ICSF-unterstützten Hardware ausgeführt werden, wenn die Ausführung im Nicht-FIPS-Modus ausgeführt wird, werden bei der Ausführung im FIPS-Modus weiterhin genutzt, mit Ausnahme der Generierung von RSA-Signaturen, die in Software ausgeführt werden müssen.

Tabelle 2. Unterschiede zwischen FIPS-Modus und Nicht-FIPS-Algorithmus-Unterstützung.

	Nicht-FIPS		**FIPS**	
Algorithm	**Schlüsselgrößen**	**Hardware**	**Schlüsselgrößen**	**Hardware**
RC2	40 und 128			
RC4	40 und 128			
DES	56	x		
TDES	168	x	168	x
AES (Advanced Encryption Standard)	128 und 256	x	128 und 256	x
MD5	48			
SHA-1	160	x	160	x

Tabelle 2. Unterschiede zwischen FIPS-Modus und Nicht-FIPS-Algorithmus-Unterstützung. (Forts.)				
	Nicht-FIPS		**FIPS**	
Algorithm	**Schlüsselgrößen**	**Hardware**	**Schlüsselgrößen**	**Hardware**
SHA-2	224, 256, 384 und 512	x	224, 256, 384 und 512	x
RSA	512-4096	x	1024-4096	x
DSA	512-1024		1024	
DH	512-2048		2048	

Im FIPS-Modus kann System SSL nur Zertifikate verwenden, die die in Tabelle 1 gezeigten Algorithmen und Schlüsselgrößen verwenden. Wenn bei der Zertifikatsvalidierung von X.509 ein Algorithmus festgestellt wird, der mit dem FIPS-Modus nicht kompatibel ist, kann das Zertifikat nicht verwendet werden und wird als ungültig behandelt.

Informationen zu Anwendungen von IBM MQ-Klassen, die den Clientmodus in WebSphere Application Server verwenden, finden Sie in FIPS-Unterstützung.

Informationen zur Konfiguration des System-SSL-Moduls finden Sie unter System SSL Module Verification Setup.

Zugehörige Verweise

„Federal Information Processing Standards” auf Seite 22
Die US-Regierung produziert technische Beratung zu IT-Systemen und Sicherheit, einschließlich der Datenverschlüsselung. Das National Institute for Standards and Technology (NIST) ist ein wichtiges Gremium, das sich mit IT-Systemen und der Sicherheit befasst. NIST erstellt Empfehlungen und Standards, einschließlich der Federal Information Processing Standards (FIPS).

Multi

TLS-Konfiguration Ihres Warteschlangenmanagers mit `mqcertck` prüfen

Bei dem Befehl **MQCERTCK** handelt es sich um ein Tool, mit dem allgemeine Fehler in der TLS-Konfiguration Ihres Warteschlangenmanagers gesucht werden können und das einige Vorschläge zur Problembehebung bereitstellt.

Einführung

Der Befehl **`mqcertck`** überprüft Folgendes:

- Vorhandensein und Berechtigungen des Schlüsselrepositorys des Warteschlangenmanagers, auf das im Warteschlangenmanager**SSLKEYR**-Attribut verwiesen wird.
- Vorhandensein und Gültigkeit des Zertifikats für den Warteschlangenmanager, auf das im Warteschlangenmanager**CERTLABL**-Attribut verwiesen wird.
- Vorhandensein und Gültigkeit aller Zertifikate, auf die in den **CERTLABL** -Attributen des TLS-fähigen Kanals verwiesen wird.
- Schlüsselrepository und Zertifikate der Clientanwendungen, einschließlich der Überprüfung, ob die Zertifikat mit dem Warteschlangenmanager berechtigt sind.

Anmerkung: Der Befehl **`mqcertck`** ist unter z/OS oder IBM i nicht verfügbar.

Verwendung

Zur Verwendung des Befehls **`mqcertck`** führen Sie den Befehl `mqcertck` zusammen mit den erforderlichen Parametern sowie gegebenenfalls allen optional erforderlichen Parametern aus einer Befehlszeile aus.

Unter mqcertck finden Sie eine Beschreibung des Befehls und der Parameters, die der Befehl verwendet.

Beispiel

Sie haben soeben Ihren Warteschlangenmanager eingerichtet`QM1`, um TLS-Verbindungen von Clients zuzulassen, die sich mit dem SVRCONN-Kanal Ihres Warteschlangenmanagers verbinden.

Sie verwenden die Funktion für mehrere Zertifikate, und somit haben sowohl Ihr Warteschlangenmanager als auch Ihr Kanal ein in ihren **CERTLABL**Attributen angegebenes Zertifikatslabel. Beim Erstellen des Kanals haben Sie im Attribut **CERTLABL** des Kanals einen Fehler gemacht. Wenn also ein Client versucht, eine Verbindung herzustellen, gibt der Warteschlangenmanager den Rückkehrcode 2393 von MQRC_SSL_INITIALIZATION_ERROR zurück.

Vor dem Aktivieren des Warteschlangenmanagers überprüfen Sie mit dem Befehl **mqcertck** die TLS-Konfiguration des Warteschlangenmanagers.

Sie führen den Befehl `mqcertck QM1` aus und empfangen die folgende Ausgabe:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
+-----------------------------------------------------------
| IBM MQ TLS Configuration Test tool
+-----------------------------------------------------------
| Problem identified:
|  No certificate could be found for the channel
|  MQCERTCK.CHANNEL
|  This tool looked in the Queue Manager's key repository
|  located at: 'C:\MQ Data\qmgrs\QM1\ssl\key.kdb'
|  for a certificate with label 'chacert',
|  which is the certificate specified in the channel's
|  CERTLABL attribute, but was unable to find one.
|
| Possible resolution:
|  A valid certificate with the label chacert
|  needs to be added to the key repository.
|
|  Alternatively, alter the channel definition to remove
|  the CERTLABL value. This can be done by executing the
|  following command in runmqsc:
|      ALTER CHANNEL(<Name>) CHLTYPE(<TYPE>) CERTLABL(' ')
+-----------------------------------------------------------
| mqcertck has ended. See above for any problems found.
| If there are problems then resolve these and run this
| tool again.
|
+-----------------------------------------------------------
```

In dieser Ausgabe werden Sie aufgefordert, Ihre Kanaldefinition für den Serververbindungskanal MQCERTCK.CHANNEL zu prüfen. Hier sehen Sie den Fehler, den Sie gemacht haben, und Sie können den Fehler beheben, bevor Sie den Befehl `mqcertck` erneut ausführen, um zu überprüfen, ob Sie das Problem behoben haben.

Clientverbindungen prüfen

Mit dem Befehl **mqcertck** können Clientschlüsselrepositorys sowie die TLS-Konfiguration des Warteschlangenmanagers überprüft werden. Dazu benötigt **mqcertck** Zugriff aus das Schlüsselrepositorys des Clients aus dem System, auf dem der Warteschlangenmanager ausgeführt.

Wenn Sie beim Ausführen des Befehls **mqcertck** dem Parameter **-clientkeyr** die Position des Clientschlüsselrepositorys (ausschließlich der Erweiterung) angeben, überprüft **mqcertck** dieses Schlüsselrepository anhand des Warteschlangenmanagers.

Wenn Sie wissen, welchen Kanal der Client für die Verbindung mit dem Warteschlangenmanager verwenden wird, können Sie dies mit dem Flag **-clientchannel** angeben.

Wenn der Client die gegenseitige Authentifizierung für die Verbindung zum Warteschlangenmanager verwendet, können Sie dem Befehl **mqcertck** mit dem Parameter **-clientusername** oder **-clientlabel** angeben, welches Zertifikat im Clientschlüsselrepository verwendet werden soll.

Wenn Sie das Standardzertifikat verwenden und der Client-Anwendung kein Zertifikatslabel zur Verfügung stellen, können Sie**-clientusername** und die **username**Parameter verwenden, die diese Anwendung ausführen.

Während der Operation des Befehls **mqcertck** erzeugt der Befehl die Zertifikatsbezeichnung `ibmwebspheremqXXXX`, wobei XXXX der Wert ist, der im Parameter **-clientusername** übergeben wurde.

Für die vollständige Überprüfung des Clientschlüsselrepositorys erstellt der Befehl **mqcertck** eine Testverbindung mit GSKit. Dazu muss der Befehl über einen Port verfügen, an den er sich während seiner Clienttests binden kann. Der verwendete Standardport ist 5857. Wenn dieser Port jedoch bereits verwendet wird, können Sie einen anderen Port angeben, der während der Clienttests verwendet werden soll.

Anmerkung: Obwohl der Port **mqcertck** an einen Port gebunden ist, wird von **mqcertck** keine externe Kommunikation verwendet und alle Tests werden lokal ausgeführt.

SSL/TLS auf dem IBM MQ MQI client

IBM MQ unterstützt TLS auf Clients. Sie können die Verwendung von TLS auf verschiedene Arten anpassen.

IBM MQ stellt die TLS-Unterstützung für IBM MQ MQI clients auf AIX, Linux, and Windows-Systemen bereit. Wenn Sie IBM MQ classes for Javaverwenden, lesen Sie die Informationen unter Using IBM MQ classes for Java , und wenn Sie IBM MQ classes for JMSverwenden, finden Sie weitere Informationen unter Using IBM MQ classes for JMS. Der restliche Teil dieses Abschnitts trifft auf Java- oder JMS-Umgebungen nicht zu.

Sie können das Schlüsselrepository für einen IBM MQ MQI client mit dem Wert MQSSLKEYR in Ihrer IBM MQ-Clientkonfigurationsdatei oder bei einem MQCONNX-Aufruf Ihrer Anwendung angeben. Es gibt drei Optionen für die Angabe, dass ein Kanal TLS verwendet:

- Verwenden einer Kanaldefinitionstabelle
- Verwendung der SSL-Konfigurationsoptionsstruktur, MQSCO, in einem MQCONNX-Aufruf
- Active Directory verwenden (auf Windows-Systemen)

Sie können die Umgebungsvariable MQSERVER nicht verwenden, um anzugeben, dass ein Kanal TLS verwendet.

Sie können die bereits vorhandenen IBM MQ MQI client-Anwendungen auch weiterhin ohne TLS verwenden, sofern TLS nicht am anderen Kanalende angegeben ist.

Wenn Änderungen auf einem Clientsystem auf den Inhalt des TLS-Schlüsselrepositorys, die Position des TLS-Schlüsselrepositorys, die Authentifizierungsdaten oder die Verschlüsselungshardware-Parameter vorgenommen werden, müssen Sie alle TLS-Verbindungen beenden, um diese Änderungen in den Clientverbindungskanälen, die die Anwendung verwendet, um eine Verbindung zum Warteschlangenmanager herzustellen, zu berücksichtigen. Wenn alle Verbindungen beendet sind, starten Sie die TLS-Kanäle erneut. Alle neuen TLS-Einstellungen werden verwendet. Diese Einstellungen entsprechen den Einstellungen, die mit dem Befehl REFRESH SECURITY TYPE (SSL) auf WS-Managersystemen aktualisiert werden.

Wenn Ihr IBM MQ MQI client auf einem AIX, Linux, and Windows-System mit Verschlüsselungshardware ausgeführt wird, konfigurieren Sie diese Hardware mit der Umgebungsvariablen „MQSSLCRYP“. Diese Variable ist äquivalent mit dem Parameter SSLCRYP im MQSC-Befehl ALTER QMGR. Im Abschnitt ALTER QMGR finden Sie eine Beschreibung des Parameters SSLCRYP im MQSC-Befehl ALTER QMGR. Wenn Sie die GSK_PCS11-Version des Parameters SSLCRYP verwenden, muss der Kennsatz PKCS #11 vollständig in Kleinbuchstaben angegeben werden.

Das Zurücksetzen des geheimen TLS-Schlüssels wird auf IBM MQ MQI clients unterstützt. Weitere Informationen finden Sie unter „Zurücksetzen von geheimen SSL-und TLS-Schlüsseln” auf Seite 497 und „Federal Information Processing Standards (FIPS) für AIX, Linux, and Windows” auf Seite 37.

Im Abschnitt „IBM MQ MQI client-Sicherheit einrichten” auf Seite 289 finden Sie weitere Informationen zur TLS-Unterstützung für IBM MQ MQI clients.

Zugehörige Tasks

Client mit einer Konfigurationsdatei konfigurieren

Angeben, dass ein MQI-Kanal SSL/TLS verwendet
Damit TLS von einem MQI-Kanal verwendet werden kann, muss der Wert des Attributs *SSLCipherSpec* für den Clientverbindungskanal mit dem Namen einer CipherSpec übereinstimmen, die von IBM MQ auf der Clientplattform unterstützt wird.

Sie können einen Clientverbindungskanal mit einem Wert für dieses Attribut auf die folgenden Arten definieren. Sie werden in der Reihenfolge absteigender Vorrangstellung aufgelistet.

1. Wenn ein PreConnect-Exit eine Kanaldefinitionsstruktur zur Verwendung bereitstellt.

   Ein PreConnect-Exit kann den Namen einer CipherSpec im Feld *SSLCipherSpec* einer Kanaldefinitionsstruktur (MQCD) angeben. Diese Struktur wird im Feld **ppMQCDArrayPtr** der MQNXP-Exit-Parameterstruktur zurückgegeben, die vom PreConnect-Exit verwendet wird.
2. Wenn eine IBM MQ MQI client-Anwendung einen MQCONNX-Aufruf ausgibt.

   Die Anwendung kann den Namen einer CipherSpec im Feld *SSLCipherSpec* einer Kanaldefinitionsstruktur (MQCD) angeben.Auf diese Struktur wird durch die Verbindungsoptionsstruktur MQCNO verwiesen, die ein Parameter im MQCONNX-Aufruf ist.
3. Verwendung einer Clientkanaldefinitionstabelle (CCDT).

   Ein oder mehrere Einträge in einer Clientkanaldefinitionstabelle können den Namen einer CipherSpec angeben.Wenn Sie beispielsweise einen Eintrag mit dem MQSC-Befehl DEFINE CHANNEL erstellen, können Sie den Parameter SSLCIPH im Befehl verwenden, um den Namen einer CipherSpec anzugeben.
4. Active Directory unter Windows verwenden.

   Auf Windows-Systemen können Sie mit dem Steuerbefehl **setmqscp** die Definitionen für den Clientverbindungskanal in Active Directory veröffentlichen. Eine oder mehrere dieser Definitionen können den Namen einer Verschlüsselungsspezifikation (CipherSpec) angeben.

Wenn eine Clientanwendung beispielsweise eine Definition für einen Clientverbindungskanal in einer MQCD-Struktur eines MQCONNX-Aufrufs bereitstellt, wird diese Definition bevorzugt vor allen anderen Einträgen in einer Definitionstabelle für den Clientkanal verwendet, auf die der IBM MQ-Client zugreifen kann.

Sie können die Umgebungsvariable MQSERVER nicht verwenden, um die Kanaldefinition auf dem Clientende eines MQI-Kanals bereitzustellen, der TLS verwendet.

Um zu überprüfen, ob ein Clientzertifikat geflossen ist, zeigen Sie den Kanalstatus am Serverende eines Kanals für das Vorhandensein eines Parameterwerts des Peernamens an.

Zugehörige Konzepte
„CipherSpec für einen IBM MQ MQI client angeben” auf Seite 474
Sie haben drei Optionen für die Angabe eines CipherSpec für einen IBM MQ MQI client.

CipherSpecs und CipherSuites in IBM MQ
IBM MQ unterstützt TLS1.3-und TLS 1.2-CipherSpecs sowie RSA- und Diffie-Hellman-Algorithmen. Sie können jedoch veraltete CipherSpecs aktivieren, wenn dies erforderlich ist.

Weitere Informationen finden Sie unter „CipherSpecs aktivieren” auf Seite 448:

- Von IBM MQ unterstützte CipherSpecs.
- Aktivieren von veralteten CipherSpecs für SSL 3.0 und TLS 1.0.

IBM MQ unterstützt RSA und den Diffie-Hellman-Schlüsselaustausch und Authentifizierungsalgorithmen. Die Größe des Schlüssels, der während des TLS-Handshake verwendet wird, kann von dem verwendeten digitalen Zertifikat abhängig sein, aber einige CipherSpecs enthalten eine Spezifikation der Schlüsselgröße des Handshake. Größere Handshake-Schlüsselgrößen bieten eine stärkere Authentifizierung. Bei kleineren Schlüsselgrößen ist der Handshake schneller.

Zugehörige Konzepte
„CipherSpecs und CipherSuites” auf Seite 21

Kryptografische Sicherheitsprotokolle müssen sich auf die Algorithmen einigen, die von einer sicheren Verbindung verwendet werden. CipherSpecs und CipherSuites definieren bestimmte Kombinationen von Algorithmen.

NSA Suite B-Verschlüsselung in IBM MQ

In diesem Abschnitt finden Sie Informationen zur Konfiguration von IBM MQ for AIX, Linux, and Windows für die Konformität mit dem mit Suite B konformen TLS 1.2-Profil.

Im Laufe der Zeit wird die NSA Cryptography Suite B Standard aktualisiert, um neue Angriffe auf Verschlüsselungsalgorithmen und -protokolle zu widerzuspiegeln. Beispiel: Einige CipherSpecs können nicht mehr Suite B zertifiziert sein. Wenn solche Änderungen auftreten, wird IBM MQ ebenfalls aktualisiert, um den neuesten Standard zu implementieren. Daher werden nach der Anwendung der Wartung möglicherweise Änderungen im Verhalten angezeigt. In der Readme-Datei für IBM MQ wird die Version von Suite B aufgelistet, die von der jeweiligen Stufe der Produktwartung umgesetzt wird. Wenn Sie IBM MQ für die Umsetzung der Suite B-Konformität konfigurieren, lesen Sie die Readme-Datei immer, wenn Sie die Wartung anwenden möchten. Siehe Produkt-Readmes für IBM MQ, WebSphere MQund MQSeries.

Auf AIX, Linux, and Windows-Systemen kann IBM MQ so konfiguriert werden, dass es dem Suite B-konformen TLS 1.2-Profil auf den in Tabelle 1 gezeigten Sicherheitsstufen entspricht.

Tabelle 3. Suite B-Sicherheitsstufen mit erlaubten CipherSpecs und digitalen Signaturalgorithmen

Sicherheitsstufe	**Zulässige CipherSpecs**	**Zulässige digitale Signaturalgorithmen**
128-Bit	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA mit SHA-256 ECDSA mit SHA-384
192-Bit	ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA mit SHA-384
Beide [1]	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA mit SHA-256 ECDSA mit SHA-384

1. Es ist möglich, sowohl die 128-Bit-als auch die 192-Bit-Sicherheitsstufe gleichzeitig zu konfigurieren. Da die Suite B-Konfiguration die minimal zulässigen Verschlüsselungsalgorithmen bestimmt, ist die Konfiguration beider Sicherheitsstufen äquivalent zur Konfiguration nur der Sicherheitsstufe 128-Bit. Die Verschlüsselungsalgorithmen der 192-Bit-Sicherheitsstufe sind stärker als die für die 128-Bit-Sicherheitsstufe erforderlichen Mindestsicherheitsstufen, so dass sie für die 128-Bit-Sicherheitsstufe auch dann zugelassen werden, wenn die 192-Bit-Sicherheitsstufe nicht aktiviert ist.

Anmerkung: Die Namenskonventionen, die für die Sicherheitsstufe verwendet werden, stellen nicht notwendigerweise die elliptische Kurvengröße oder die Schlüsselgröße des AES-Verschlüsselungsalgorithmus dar.

CipherSpec-Konformation zu Suite B

Obwohl das Standardverhalten von IBM MQ nicht dem Suite B-Standard entspricht, kann IBM MQ so konfiguriert werden, dass es entweder einer oder beiden Sicherheitsstufen auf AIX, Linux, and Windows-Systemen entspricht. Direkt nach der erfolgreichen Konfiguration von IBM MQ für die Verwendung von Suite B führt jeder Versuch, einen Kanal für abgehende Nachrichten mit einer nicht Suite B-konformen CipherSpec zu starten, zu dem Fehler `AMQ9282`. Diese Aktivität führt auch dazu, dass der MQI-Client den Ursachencode `MQRC_CIPHER_SPEC_NOT_SUITE_B` zurückgibt. Bei dem Versuch, einen eingehenden Kanal unter Verwendung einer CipherSpec zu starten, die nicht der Suite B-Konfiguration entspricht, wird der Fehler `AMQ9616` angezeigt.

Weitere Informationen zu IBM MQ-CipherSpecs finden Sie unter „CipherSpecs aktivieren” auf Seite 448

Suite B und digitale Zertifikate

Suite B beschränkt die digitalen Signaturalgorithmen, die zum Signieren digitaler Zertifikate verwendet werden können. Suite B schränkt auch die Art des öffentlichen Schlüssels ein, den Zertifikate enthalten können. Daher muss IBM MQ für die Verwendung von Zertifikaten konfiguriert werden, deren Algorithmus für digitale Signaturen und öffentlicher Schlüsseltyp für die konfigurierten Sicherheitsstufe der Suite B des fernen Partner zulässig ist. Digitale Zertifikate, die nicht den Anforderungen der Sicherheitsstufe entsprechen, werden zurückgewiesen, und die Verbindung schlägt mit Fehler AMQ9633 oder AMQ9285 fehl.

Für die Sicherheitsstufe der 128-Bit-Suite B ist der öffentliche Schlüssel des Zertifikatsubjekt erforderlich, um entweder die elliptische NIST P-256-Kurve oder die NIST P-384-elliptische Kurve zu verwenden und entweder mit der elliptischen NIST P-256-Kurve oder mit der NIST P-384-elliptischen Kurve signiert zu werden. Auf der Sicherheitsebene der 192-Bit-Suite B ist der öffentliche Schlüssel des Zertifikatsubjekt erforderlich, um die NIST P-384-elliptische Kurve zu verwenden und mit der elliptischen NIST P-384-Kurve signiert werden zu können.

Um ein Zertifikat abzurufen, das für Suite B-konforme Operationen geeignet ist, verwenden Sie den Befehl **runmqakm** und geben Sie den Parameter **-sig_alg** an, um einen geeigneten digitalen Signaturalgorithmus anzufordern. Die Parameterwerte `EC_ecdsa_with_SHA256` und `EC_ecdsa_with_SHA384` **-sig_alg** entsprechen elliptischen Kurvenschlüsseln, die von den digitalen Signaturalgorithmen der Suite B signiert sind.

Weitere Informationen zum Befehl **runmqakm** finden Sie unter runmqckm-und runmqakm-Optionen.

Anmerkung: Die Befehle **runmqckm** und **strmqikm** unterstützen nicht die Erstellung von digitalen Zertifikaten für die Suite B-konformen Operation.

Erstellen und Anfordern von digitalen Zertifikaten

Informationen zum Erstellen eines selbst signierten digitalen Zertifikats für Suite B-Tests finden Sie in „Selbst signiertes persönliches Zertifikat unter AIX, Linux, and Windows erstellen” auf Seite 315.

Informationen zum Anfordern eines von einer Zertifizierungsstelle signierten digitalen Zertifikats für die Produktionsverwendung in Suite B finden Sie in „Persönliches Zertifikat unter AIX, Linux, and Windows anfordern” auf Seite 318.

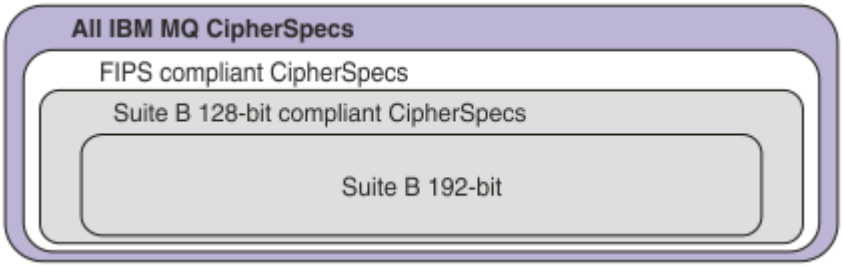
Anmerkung: Die verwendete Zertifizierungsstelle muss digitale Zertifikate generieren, die die in der IETF-RFC 6460 beschriebenen Anforderungen erfüllen.

FIPS 140-2 und Suite B

Anmerkung: Unter AIX, Linux, and Windowsstellt IBM MQ Konformität mit FIPS 140-2 über das Verschlüsselungsmodul "IBM Crypto for C" bereit. Das Zertifikat für dieses Modul wurde in den Langzeitstatus versetzt. Kunden sollten das IBM Crypto for C-Zertifikat anzeigen und sich über Empfehlungen von NIST im Klaren sein. Ein Ersatz-FIPS 140-3-Modul ist derzeit in Bearbeitung und sein Status kann angezeigt werden, indem in der NIST-CMVP-Module in der Prozesslistenach ihm gesucht wird.

Der Suite B-Standard ist konzeptionell ähnlich wie FIPS 140-2, da er die Menge aktivierter kryptografischer Algorithmen einschränkt, um ein gesichertes Sicherheitsniveau zu gewährleisten. Die derzeit unterstützten Suite B-CipherSpecs können verwendet werden, wenn IBM MQ für den FIPS 140-2-konformen Betrieb konfiguriert wurde. Daher kann IBM MQ für die gleichzeitige FIPS- und Suite B-Konformität konfiguriert werden, wofür dann beide Gruppen von Einschränkungen gelten.

Das folgende Diagramm veranschaulicht die Beziehung zwischen diesen Untergruppen:



IBM MQ für Suite B-konformen Betrieb konfigurieren

Informationen zum Konfigurieren von IBM MQ in AIX, Linux, and Windows für einen Suite B-konformen Vorgang finden Sie unter „IBM MQ für Suite B konfigurieren” auf Seite 47.

IBM MQ unterstützt nicht den Suite B-konformen Betrieb auf den IBM i- und z/OS-Plattformen. Die IBM MQ Java- und JMS-Clients unterstützen ebenfalls keinen Suite B-konforme Betrieb.

Zugehörige Konzepte

„Angeben, dass nur FIPS-zertifizierte CipherSpecs während der Ausführung auf dem MQI-Client verwendet werden” auf Seite 290
Erstellen Sie Ihre Schlüsselrepositorys mit FIPS-konformer Software und geben Sie dann an, dass der Kanal FIPS-zertifizierte CipherSpecs verwenden muss.

ALW *IBM MQ für Suite B konfigurieren*

IBM MQ kann so konfiguriert werden, dass sie in Übereinstimmung mit dem NSA Suite B-Standard auf AIX, Linux, and Windows-Plattformen ausgeführt wird.

Suite B beschränkt die Gruppe aktivierter Verschlüsselungsalgorithmen, um eine sichere Sicherheitsstufe zu gewährleisten. IBM MQ kann so konfiguriert werden, dass es in Übereinstimmung mit Suite B ausgeführt wird, um ein erhöhtes Sicherheitsniveau bereitzustellen. Weitere Informationen zu Suite B finden Sie in „National Security Agency (NSA) Suite B Cryptography” auf Seite 23. Weitere Informationen zur Suite B-Konfiguration und deren Auswirkungen auf TLS-Kanäle finden Sie in „NSA Suite B-Verschlüsselung in IBM MQ” auf Seite 45.

Warteschlangenmanager

Für einen Warteschlangenmanager verwenden Sie den Befehl **ALTER QMGR** mit dem Parameter **SUITEB** , um die entsprechenden Werte für Ihre erforderliche Sicherheitsstufe festzulegen. Weitere Informationen finden Sie unter ALTER QMGR .

Sie können auch den PCF-Befehl **MQCMD_CHANGE_Q_MGR** mit dem Parameter **MQIA_SUITE_B_STRENGTH** verwenden, um den Warteschlangenmanager für Suite B-konforme Operationen zu konfigurieren.

Anmerkung: Wenn Sie die Einstellungen für Suite B eines Warteschlangenmanagers ändern, müssen Sie den MQXR-Service erneut starten, damit diese Einstellungen wirksam werden.

MQI-Client

Standardmäßig erzwingen MQI-Clients die Suite B-Konformität nicht. Sie können den MQI-Client für die Suite B-Konformität aktivieren, indem Sie eine der folgenden Optionen ausführen:

1. Durch Festlegen des Felds **EncryptionPolicySuiteB** in der MQSCo-Struktur in einem MQCONNX-Aufruf auf einen oder mehrere der folgenden Werte:
   - MQ_SUITE_B_NONE
   - MQ_SUITE_B_128_BIT
   - MQ_SUITE_B_192_BIT

Die Verwendung von MQ_SUITE_B_NONE mit einem anderen Wert ist ungültig.

2. Indem Sie die Umgebungsvariable MQSUITEB auf einen oder mehrere der folgenden Werte setzen:
   - KEINE
   - 128_BIT
   - 192_BIT

   Sie können mehrere Werte in einer durch Kommas getrennten Liste angeben. Die Verwendung des Werts NONE mit einem beliebigen anderen Wert ist ungültig.
3. Durch Festlegen des Attributs **EncryptionPolicySuiteB** in der SSL-Zeilengruppe der MQI-Clientkonfigurationsdatei auf einen oder mehrere der folgenden Werte:
   - KEINE
   - 128_BIT
   - 192_BIT

   Sie können mehrere Werte in einer durch Kommas getrennten Liste angeben. Die Verwendung von NONE mit einem anderen Wert ist ungültig.

Anmerkung: Die MQI-Clienteinstellungen werden in der Reihenfolge ihrer Priorität aufgelistet. Die MSCO-Struktur für den MQCONNX-Aufruf überschreibt die Einstellung in der Umgebungsvariablen MQSUITEB, die das Attribut in der SSL-Zeilengruppe überschreibt.

Ausführliche Informationen zur MQSCO-Struktur finden Sie im Abschnitt MQSCO-SSL-Konfigurationsoptionen .

Weitere Informationen zur Verwendung von Suite B in der Clientkonfigurationsdatei finden Sie in der SSL-Zeilengruppe der Clientkonfigurationsdatei .

Weitere Informationen zur Verwendung der Umgebungsvariablen MQSUITEB finden Sie unter Beschreibungen der Umgebungsvariablen.

.NET

Für nicht verwaltete .NET -Clients gibt die Eigenschaft **MQC.ENCRYPTION_POLICY_SUITE_B** den Typ der erforderlichen Suite B-Sicherheit an.

Informationen zur Verwendung von Suite B in IBM MQ classes for .NET finden Sie im Abschnitt MQEnvironment .NET-Klasse.

AMQP

Die Attributeinstellungen von Suite B für einen Warteschlangenmanager gelten für AMQP-Kanäle dieses Warteschlangenmanagers. Wenn Sie die Einstellungen des Warteschlangenmanagers Suite B ändern, müssen Sie den AMQP-Service erneut starten, damit die Änderungen wirksam werden.

Zertifikatsprüfrichtlinien in IBM MQ

Die Zertifikatvalidierungs-Richtlinie bestimmt, wie streng die Validierung der Zertifikatskette den Branchensicherheitsstandards entspricht.

Die Richtlinie für die Zertifikatsprüfung hängt wie folgt von der Plattform und der Umgebung ab:

- Für Java- und JMS-Anwendungen auf allen Plattformen hängt die Zertifikatsprüfrichtlinie von der SSE-Komponente der Java Runtime Environment ab. Weitere Informationen zur Validierungsrichtlinie für Zertifikate finden Sie in der Dokumentation zu Ihrer JRE.
- Für IBM i-Systeme hängt die Zertifikatsprüfrichtlinie von der Secure Sockets-Bibliothek ab, die vom Betriebssystem bereitgestellt wird. Weitere Informationen zur Gültigkeitsprüfungspolitik für Zertifikate finden Sie in der Dokumentation zum Betriebssystem.

- Für z/OS-Systeme hängt die Zertifikatsprüfrichtlinie von der System SSL-Komponente ab, die vom Betriebssystem bereitgestellt wird. Weitere Informationen zur Gültigkeitsprüfungspolitik für Zertifikate finden Sie in der Dokumentation zum Betriebssystem.
- Für Systeme mit AIX, Linux, and Windows wird die Zertifikatsprüfrichtlinie mit dem GSKit bereitgestellt und kann konfiguriert werden. Es werden zwei unterschiedliche Validierungsrichtlinien für Zertifikate unterstützt:
  - Eine traditionelle Zertifikatvalidierungsrichtlinie, die für die maximale Abwärtskompatibilität und die Interoperabilität mit alten digitalen Zertifikaten verwendet wird, die nicht den aktuellen IETF-Zertifikatsprüfstandards entsprechen. Diese Richtlinie wird als Grundrichtlinie bezeichnet.
  - Eine strenge, standardkonforme Zertifikatvalidierungsrichtlinie, die den Standard RFC 5280 erzwingt. Diese Richtlinie wird als Standardrichtlinie bezeichnet.

Weitere Informationen zum Konfigurieren der Zertifikatsprüfrichtlinie unter AIX, Linux, and Windows finden Sie im Abschnitt „Zertifikatsprüfrichtlinien in IBM MQ konfigurieren” auf Seite 49. Weitere Informationen zu den Unterschieden zwischen der Zertifikatsprüfung mit Basis- und Standardrichtlinien finden Sie unter Zertifikatsvalidierung und Entwicklung von Trust-Richtlinien auf AIX, Linux, and Windows.

Zertifikatsprüfrichtlinien in IBM MQ konfigurieren

Sie können angeben, welche TLS-Zertifikatsprüfungs-Policy verwendet wird, um digitale Zertifikate zu validieren, die von fernen Partnersystemen auf vier Arten empfangen werden.

Auf dem Warteschlangenmanager kann die Zertifikatsprüfungs-Policy wie folgt festgelegt werden:

- Verwenden Sie das WS-Managerattribut *CERTVPOL*. Weitere Informationen zum Festlegen dieses Attributs finden Sie in ALTER QMGR .

Auf dem Client gibt es verschiedene Methoden, mit denen die Validierungsrichtlinie für Zertifikate festgelegt werden kann. Wenn mehr als eine Methode zum Festlegen der Richtlinie verwendet wird, verwendet der Client die Einstellungen in der folgenden Prioritätsreihenfolge:

1. Verwenden Sie das Feld *CertificateValPolicy* in der MQSCO-Clientstruktur. Weitere Informationen zur Verwendung dieses Felds finden Sie im Abschnitt MQSCO-SSL-Konfigurationsoptionen .
2. Verwenden Sie die Clientumgebungsvariable *MQCERTVPOL*. Weitere Informationen zur Verwendung dieser Variablen finden Sie in MQCERTVPOL .
3. Verwenden Sie die Einstellung des Parameters für die Optimierung von Client-SSL-Einstellungen, *CertificateValPolicy* Weitere Informationen zur Verwendung dieser Einstellung finden Sie in der SSL-Zeilengruppe der Clientkonfigurationsdatei .

Weitere Informationen zu Zertifikaten für Zertifikatvalidierungen finden Sie in „Zertifikatsprüfrichtlinien in IBM MQ” auf Seite 48.

Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ

Dieser Abschnitt enthält Informationen dazu, wie Sie die richtigen CipherSpecs und digitalen Zertifikate für Ihre Sicherheitsrichtlinie auswählen. Dazu wird die Beziehung zwischen CipherSpecs und digitalen Zertifikaten in IBM MQ erläutert.

Nur eine Untergruppe der unterstützten CipherSpecs kann mit allen unterstützten Typen von digitalen Zertifikaten verwendet werden. Es ist daher notwendig, eine geeignete CipherSpec für Ihr digitales Zertifikat zu wählen. Wenn die Sicherheitsrichtlinie Ihres Unternehmens die Verwendung einer bestimmten CipherSpec-Spezifikation erfordert, müssen Sie außerdem ein entsprechendes digitales Zertifikat für diese CipherSpec erwerben.

Digitales MD5-Signaturalgorithmus und TLS 1.2

Digitale Zertifikate, die mit dem MD5-Algorithmus signiert sind, werden zurückgewiesen, wenn das TLS 1.2-Protokoll verwendet wird. Dies liegt daran, dass der MD5-Algorithmus jetzt von vielen kryptografischen Analysten als schwach angesehen wird und die Verwendung im Allgemeinen nicht geworben wird. Wenn Sie neuere CipherSpecs auf der Basis des TLS 1.2-Protokolls verwenden möchten, müssen Sie sicherstellen, dass die digitalen Zertifikate den MD5-Algorithmus nicht in ihren digitalen Signaturen

verwenden. Ältere CipherSpecs, die die TLS 1.0-Protokolle verwenden, unterliegen dieser Einschränkung nicht und können weiterhin Zertifikate mit digitalen MD5-Signaturen verwenden.

Um den Algorithmus für digitale Signatur für ein bestimmtes Zertifikat anzuzeigen, können Sie den Befehl **runmqakm** verwenden:

```
runmqakm -cert -details -db key.kdb -pw password -label cert_label
```

Hierbei steht *cert_label* für den Zertifikatskennsatz des Algorithmus für digitale Signatur, der angezeigt werden soll. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

Anmerkung: Obwohl die grafische Benutzerschnittstelle **runmqckm** (iKeycmd) und **strmqikm** (iKeyman) verwendet werden kann, um eine Auswahl digitaler Signaturalgorithmen anzuzeigen, stellt das Tool **runmqakm** einen größeren Bereich bereit.

Bei der Ausführung des Befehls **runmqakm** wird die Ausgabe mit der Verwendung des angegebenen Signaturalgorithmus ausgegeben:

```
Label : ibmmqexample
Key Size : 1024
Version : X509 V3
Serial : 4e4e93f1
Issuer : CN=Old Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : August 19, 2011 5:48:49 PM GMT+01:00
Not After : August 18, 2012 5:48:49 PM GMT+01:00
Public Key
    30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
    05 00 03 81 8D 00 30 81 89 02 81 81 00 98 5A 7A
    F0 18 21 EE E4 8A 6E DE C8 01 4B 3A 1E 41 90 3D
    CE 01 3F E6 32 30 6C 23 59 F0 FE 78 6D C2 80 EF
    BC 83 54 7A EB 60 80 62 6B F1 52 FE 51 9D C1 61
    80 A5 1C D4 F0 76 C7 15 6D 1F 0D 4D 31 3E DC C6
    A9 20 84 6E 14 A1 46 7D 4C F5 79 4D 37 54 0A 3B
    A9 74 ED E7 8B 0F 80 31 63 1A 0B 20 A5 99 EE 0A
    30 A6 B6 8F 03 97 F6 99 DB 6A 58 89 7F 27 34 DE
    55 08 29 D8 A9 6B 46 E6 02 17 C3 13 D3 02 03 01
    00 01
Public Key Type : RSA (1.2.840.113549.1.1.1)
Fingerprint : SHA1 :
    09 4E 4F F2 1B CB C1 F4 4F 15 C9 2A F7 32 0A 82
    DA 45 92 9F
Fingerprint : MD5 :
    44 54 81 7C 58 68 08 3A 5D 75 96 40 D5 8C 7A CB
Fingerprint : SHA256 :
    3B 47 C6 E7 7B B0 FF 85 34 E7 48 BE 11 F2 D4 35
    B7 9A 79 53 2B 07 F5 E7 65 E8 F7 84 E0 2E 82 55
Signature Algorithm : MD5WithRSASignature (1.2.840.113549.1.1.4)
Value
    3B B9 56 E6 F2 77 94 69 5B 3F 17 EA 7B 19 D0 A2
    D7 10 38 F1 88 A4 44 1B 92 35 6F 3B ED 99 9B 3A
    A5 A4 FC 72 25 5A A9 E3 B1 96 88 FC 1E 9F 9B F1
    C5 E8 8E CF C4 8F 48 7B 0E A6 BB 13 AE 2B BD D8
    63 2C 03 38 EF DC 01 E1 1F 7A 6F FB 2F 65 74 D0
    FD 99 94 BA B2 3A D5 B4 89 6C C1 2B 43 6D E2 39
    66 6A 65 CB C3 C4 E2 CC F5 49 39 A3 8B 93 5A DD
    B0 21 0B A8 B2 59 5B 24 59 50 44 89 DC 78 19 51
Trust Status : Enabled
```

Die Zeile `Signature Algorithm` zeigt, dass der `MD5WithRSASignature` -Algorithmus verwendet wird. Dieser Algorithmus basiert auf MD5, so dass dieses digitale Zertifikat nicht mit den TLS 1.2-CipherSpecs verwendet werden kann.

Interoperabilität von Elliptic Curve und RSA CipherSpecs

V 9.2.0 Es können nicht alle CipherSpecs mit allen digitalen Zertifikaten verwendet werden. CipherSpecs werden durch das Namenspräfix CipherSpec angegeben. Jeder Typ von CipherSpec legt unterschiedliche Einschränkungen für den Typ des verwendbaren digitalen Zertifikats fest. Diese Einschränkungen gelten für alle TLS-Verbindungen von IBM MQ, sind jedoch besonders für die Benutzer von Elliptic Curve Cryptography relevant.

In der folgenden Tabelle sind die Beziehungen zwischen CipherSpecs und digitalen Zertifikaten zusammengefasst:

Tabelle 4. Beziehungen zwischen CipherSpecs und digitalen Zertifikaten

Typ	Präfix für CipherSpec-Name	Beschreibung	Erforderlicher öffentlicher Schlüsseltyp	Verschlüsselungsalgorithmus für digitale Signatur	Geheime Schlüsselniederlassmethode
1	ECDHE_ECDSA_	CipherSpecs, die Elliptic Curve Public Keys, Elliptic Curve Secret Keys, und Elliptic Curve digitale Signaturalgorithmen verwenden.	Elliptische Kurve	ECDSA	ECDHE
2	ECDHE_RSA_	CipherSpecs, die RSA-Public-Keys, Elliptic Curve-Secret-Schlüssel und digitale RSA-Signaturalgorithmen verwenden.	RSA	RSA	ECDHE
V 9.2.0 3	(Alle TLS 1.3 CipherSpecs)	CipherSpecs , die öffentliche Elliptic Curve-oder RSA-Schlüssel, geheime Elliptic Curve-Schlüssel und digitale Elliptic Curve- oder RSA-Signaturalgorithmen verwenden.	Elliptic Curve oder RSA	ECDSA oder RSA	ECDHE oder RSA
4	(Alle anderen)	CipherSpecs, die öffentliche RSA-Schlüssel und digitale RSA-Signaturalgorithmen verwenden.	RSA	RSA	RSA

Anmerkung: CipherSpecs des Typs 1 und 2 werden von IBM MQ-Warteschlangenmanagern und MQI-Clients unter IBM i nicht unterstützt.

In der erforderlichen Spalte für den öffentlichen Schlüsseltyp wird der Typ des öffentlichen Schlüssels angezeigt, den das persönliche Zertifikat bei der Verwendung jedes Typs von CipherSpec haben muss. Das persönliche Zertifikat ist das Zertifikat der Ententität, das den WS-Manager oder Client an seinen fernen Partner identifiziert.

Sie müssen sicherstellen, dass das in der Zertifikatsbezeichnung genannte Zertifikat für den Kanal CipherSpec geeignet ist. Wenn Sie also einen Kanal mit einer CipherSpec konfigurieren, für die ein EC-Zertifikat (Elliptic Curve) erforderlich ist, können Sie kein RSA-Zertifikat in der Zertifikatsbezeichnung angeben. Wenn Sie einen Kanal mit einer CipherSpec konfigurieren, für die ein RSA-Zertifikat erforderlich ist, können Sie kein EC-Zertifikat in der Zertifikatsbezeichnung angeben.

Vorausgesetzt, dass IBM MQ richtig konfiguriert ist, können Sie Folgendes verwenden:

- Ein einzelner WS-Manager mit einer Mischung aus RSA und EC-Zertifikaten.
- Unterschiedliche Kanäle auf demselben Warteschlangenmanager, die entweder ein RSA-oder ein EC-Zertifikat verwenden.

Der Verschlüsselungsalgorithmus der digitalen Signatur bezieht sich auf den Verschlüsselungsalgorithmus, der zur Validierung des Peers verwendet wird. Der Verschlüsselungsalgorithmus wird zusammen mit einem Hash-Algorithmus wie MD5, SHA-1 oder SHA-256 verwendet, um die digitale Signatur zu berechnen. Es gibt verschiedene digitale Signaturalgorithmen, die z. B. RSA mit MD5 oder ECDSA mit SHA-256 verwendet werden können. In der Tabelle bezieht sich ECDSA auf die Gruppe der digitalen Signaturalgorithmen, die ECDSA verwenden; RSA bezieht sich auf die Gruppe digitaler Signaturalgorithmen, die RSA verwenden. Jeder unterstützte digitale Signaturalgorithmus in der Gruppe kann verwendet werden, vorausgesetzt, er basiert auf dem angegebenen Verschlüsselungsalgorithmus.

CipherSpecs vom Typ 1 setzen voraus, dass das persönliche Zertifikat einen öffentlichen Öffentlichen Schlüssel (Elliptic Curve Public Key) aufweisen muss. Wenn diese CipherSpecs verwendet werden, wird mit Elliptic Curve Diffie Hellman Ephemeral key agreement der geheime Schlüssel für die Verbindung hergestellt.

CipherSpecs vom Typ 2 setzen voraus, dass das persönliche Zertifikat einen öffentlichen RSA-Schlüssel hat. Wenn diese CipherSpecs verwendet werden, wird mit Elliptic Curve Diffie Hellman Ephemeral key agreement der geheime Schlüssel für die Verbindung hergestellt.

CipherSpecs vom Typ 3 setzen voraus, dass das persönliche Zertifikat einen öffentlichen RSA-Schlüssel aufweisen muss. Wenn diese CipherSpecs verwendet werden, wird der geheime Schlüssel für die Verbindung mit einem RSA-Schlüsselaustausch aufgebaut.

Diese Liste der Einschränkungen ist nicht erschöpfend: Je nach Konfiguration kann es zusätzliche Einschränkungen geben, die weitere Auswirkungen auf die Interaktivität haben können. Wenn IBM MQ beispielsweise so konfiguriert ist, dass es mit FIPS 140-2 oder NSA Suite B-Standards konform ist, werden dadurch auch die zulässigen Konfigurationen eingeschränkt. Weitere Informationen finden Sie im folgenden Abschnitt.

Wenn Sie verschiedene CipherSpec-Typen in demselben Warteschlangenmanager oder in derselben Clientanwendung verwenden müssen, konfigurieren Sie eine entsprechende Zertifikatsbezeichnung und die CipherSpec-Kombination in der Clientdefinition.

Die drei Typen von CipherSpec sind nicht direkt interaktiv: Dies ist eine Einschränkung der aktuellen TLS-Standards. Angenommen, Sie haben die CipherSpec ECDHE_ECDSA_AES_128_CBC_SHA256 für einen Empfängerkanal mit dem Namen TO.QM1 auf einem WS-Manager mit dem Namen QM1sollte der Empfänger über ein persönliches Zertifikat mit einem Elliptic Curve-Schlüssel und einer ECDSA-basierten digitalen Signatur verfügen. Wenn der Empfängerkanal diese Anforderungen nicht erfüllt, kann der Kanal nicht gestartet werden.

Andere Kanäle, die mit WS-Manager QM1 verbunden sind, können andere CipherSpecs verwenden, sofern jeder Kanal ein Zertifikat des korrekten Typs für die CipherSpec dieses Kanals verwendet. Angenommen, QM1 verwendet einen Senderkanal mit dem Namen TO.QM2, um Nachrichten an einen anderen WS-Manager mit dem Namen QM2 zu senden. Der Kanal TO.QM2 könnte den Typ 3 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256 verwenden, vorausgesetzt, beide Enden des Kanals verwenden Zertifikate, die RSA-Public-Keys enthalten. Das Kanalattribut für das Zertifikatkennsatz kann verwendet werden, um ein anderes Zertifikat für jeden Kanal zu konfigurieren.

Berücksichtigen Sie bei der Planung Ihrer IBM MQ-Netze sorgfältig, welche Kanäle TLS benötigen, und stellen Sie sicher, dass der Typ der Zertifikate, die für jeden Kanal verwendet werden, für die Verwendung mit der CipherSpec auf diesem Kanal geeignet ist.

Zum Anzeigen des Algorithmus für digitale Signatur und des öffentlichen Schlüsseltyps für ein digitales Zertifikat können Sie den Befehl **runmqakm** verwenden:

```
runmqakm -cert -details -db key.kdb -pw password -label cert_label
```

Dabei steht *cert_label* für die Bezeichnung des Zertifikats, dessen digitaler Signaturalgorithmus Sie anzeigen müssen. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

Bei der Ausführung des Befehls **runmqakm** wird die Ausgabe mit dem Typ "Public Key" ausgegeben:

```
Label : ibmmqexample
Key Size : 384
Version : X509 V3
Serial : 9ad5eeef5d756f41
Issuer : CN=Example Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : 21 August 2011 13:10:24 GMT+01:00
Not After : 21 August 2012 13:10:24 GMT+01:00
Public Key
    30 76 30 10 06 07 2A 86 48 CE 3D 02 01 06 05 2B
    81 04 00 22 03 62 00 04 3E 6F A9 06 B6 C3 A0 11
    F8 D6 22 78 FE EF 0A FE 34 52 C0 8E AB 5E 81 73
    D0 97 3B AB D6 80 08 E7 31 E9 18 3F 6B DE 06 A7
```

```
    15 D6 9D 5B 6F 56 3B 7F 72 BB 6F 1E C9 45 1C 46
    60 BE F2 DC 1B AD AC EC 64 4C 0E 06 65 6E ED 93
    B8 F5 95 E0 F9 2A 05 D6 21 02 BD FB 06 63 A1 CC
    66 C6 8A 0A 5C 3F F7 D3
Public Key Type : EC_ecPublicKey (1.2.840.10045.2.1)
Fingerprint : SHA1 :
    3C 34 58 04 5B 63 5F 5C C9 7A E7 67 08 2B 84 43
    3D 43 7A 79
Fingerprint : MD5 :
    49 13 13 E1 B2 AC 18 9A 31 41 DC 8C B4 D6 06 68
Fingerprint : SHA256 :
    6F 76 78 68 F3 70 F1 53 CE 39 31 D9 05 C5 C5 9F
    F2 B8 EE 21 49 16 1D 90 64 6D AC EB 0C A7 74 17
Signature Algorithm : EC_ecdsa_with_SHA384 (1.2.840.10045.4.3.3)
Value
    30 65 02 30 0A B0 2F 72 39 9E 24 5A 22 FE AC 95
    0D 0C 6D 6C 2F B3 E7 81 F6 C1 36 1B 9A B0 6F 07
    59 2A A1 4C 02 13 7E DD 06 D6 FE 4B E4 03 BC B1
    AC 49 54 1E 02 31 00 90 0E 46 2B 04 37 EE 2C 5F
    1B 9C 69 E5 99 60 84 84 10 71 1A DA 63 88 33 E2
    22 CC E6 1A 4E F4 61 CC 51 F9 EE A0 8E F4 DC B5
    0B B9 72 58 C3 C7 A4
Trust Status : Enabled
```

Die Linie 'Public Key Type' (Öffentlicher Schlüssel) zeigt in diesem Fall an, dass das Zertifikat einen öffentlichen Elliptic Curve-Schlüssel hat. Die Signaturalgorithmuslinie in diesem Fall zeigt an, dass der Algorithmus EC_ecdsa_with_SHA384 im Gebrauch ist: Dies basiert auf dem ECDSA-Algorithmus. Dieses Zertifikat ist daher nur für die Verwendung mit Typ 1 CipherSpecs geeignet.

Sie können auch den Befehl **runmqckm** mit den gleichen Parametern verwenden. Außerdem kann die grafische Benutzerschnittstelle **strmqikm** verwendet werden, um digitale Signaturalgorithmen anzuzeigen, wenn Sie das Schlüsselrepository öffnen und doppelt auf die Bezeichnung des Zertifikats klicken. Sie sollten jedoch das Tool **runmqakm** verwenden, um digitale Zertifikate anzuzeigen, da es einen größeren Bereich von Algorithmen unterstützt.

TLS 1.3 CipherSpecs

V 9.2.0

TLS 1.3 CipherSpecs unterstützen sowohl ECDSA-als auch RSA-Zertifikate.

Elliptic Curve CipherSpecs und NSA Suite B

Wenn IBM MQ für die Konformität mit dem Suite B-konformen TLS 1.2-Profil konfiguriert ist, sind die zulässigen CipherSpecs und Algorithmen für digitale Signaturen wie in „NSA Suite B-Verschlüsselung in IBM MQ” auf Seite 45 beschrieben eingeschränkt. Darüber hinaus wird der Bereich der zulässigen Elliptic Curve-Schlüssel entsprechend der konfigurierten Sicherheitsstufen reduziert.

Auf der 128-Bit-Suite B ist der öffentliche Schlüssel des Zertifikatsubjektschlüssels erforderlich, um entweder die NIST P-256-oder NIST P-384-elliptische Kurve zu verwenden und entweder mit der NIST P-256-elliptischen Kurve oder mit der NIST P-384-elliptischen Kurve signiert zu werden. Der Befehl **runmqakm** kann verwendet werden, um digitale Zertifikate für diese Sicherheitsstufe mit dem Parameter -sig_alg von EC_ecdsa_with_SHA256oder EC_ecdsa_with_SHA384anzufordern.

Auf der Ebene der 192-Bit-Suite B ist der öffentliche Schlüssel des Zertifikatsubjektschlüssels erforderlich, um die NIST P-384-elliptische Kurve zu verwenden und mit der elliptischen NIST P-384-Kurve signiert werden zu können. Der Befehl **runmqakm** kann verwendet werden, um digitale Zertifikate für diese Sicherheitsstufe mit einem Parameter -sig_alg von EC_ecdsa_with_SHA384anzufordern.

Die unterstützten NIST-Elliptic-Kurven lauten wie folgt:

Tabelle 5. Unterstützte NIST-Elliptische Kurven

NIST FIPS 186-3-Kurvenname	RFC 4492-Kurvenname	Elliptische Kurvenschlüsselgröße (Bit)
P-256	secp256r1	256

Tabelle 5. Unterstützte NIST-Elliptische Kurven (Forts.)		
NIST FIPS 186-3-Kurvenname	**RFC 4492-Kurvenname**	**Elliptische Kurvenschlüsselgröße (Bit)**
P-384	secp384r1	384
P-521	secp521r1	521

Anmerkung: Die elliptische NIST P-521-Kurve kann nicht für die Suite B-konforme Operation verwendet werden.

Zugehörige Konzepte

„CipherSpecs aktivieren" auf Seite 448
Aktivieren Sie eine CipherSpec mit dem Parameter **SSLCIPH** im MQSC-Befehl **DEFINE CHANNEL** oder **ALTER CHANNEL**.

„Angeben, dass nur FIPS-zertifizierte CipherSpecs während der Ausführung auf dem MQI-Client verwendet werden" auf Seite 290
Erstellen Sie Ihre Schlüsselrepositorys mit FIPS-konformer Software und geben Sie dann an, dass der Kanal FIPS-zertifizierte CipherSpecs verwenden muss.

„NSA Suite B-Verschlüsselung in IBM MQ" auf Seite 45
In diesem Abschnitt finden Sie Informationen zur Konfiguration von IBM MQ for AIX, Linux, and Windows für die Konformität mit dem mit Suite B konformen TLS 1.2-Profil.

„National Security Agency (NSA) Suite B Cryptography" auf Seite 23
Die Regierung der Vereinigten Staaten von Amerika erstellt technische Beratung zu IT-Systemen und Sicherheit, einschließlich der Datenverschlüsselung. Die US National Security Agency (NSA) empfiehlt eine Reihe interoperabler kryptographischer Algorithmen in ihrem Suite B Standard.

Kanalauthentifizierungssätze

Die Zugriffsberechtigungen zum Herstellen von Systemverbindungen auf Kanalebene können mithilfe von Kanalauthentifizierungsdatensätzen gezielter gesteuert werden.

Möglicherweise stellen Sie fest, dass Clients versuchen, unter einer aus Leerzeichen bestehenden Benutzer-ID oder einer allgemeinen Benutzer-ID eine Verbindung mit Ihrem Warteschlangenmanager herzustellen, die es den Clients ermöglichen würde, unerwünschte Aktionen auszuführen. Sie können den Zugriff dieser Clients mithilfe von Kanalauthentifizierungssätzen blockieren. In einem anderen Fall bestätigt ein Client möglicherweise eine Benutzer-ID, die auf der Clientplattform gültig ist, aber auf der Serverplattform unbekannt ist oder ein ungültiges Format hat. Über einen Kanalauthentifizierungssatz können Sie die betreffende Benutzer-ID einer gültigen Benutzer-ID zuordnen.

Sie stellen möglicherweise fest, dass sich eine Clientanwendung, die eine Verbindung mit Ihrem Warteschlangenmanager herstellt, auf irgendeine Weise schädlich verhält. Um den Server vor Problemen zu schützen, die durch diese Anwendung verursacht werden können, muss die Clientanwendung über ihre IP-Adresse vorübergehend blockiert werden, bis die Firewallregeln aktualisiert wurden oder die Anwendung korrigiert wurde. Mithilfe eines Kanalauthentifizierungssatzes können Sie die IP-Adresse, mit der die Clientanwendung die Verbindung herstellt, blockieren.

Wenn Sie ein Verwaltungstool, z. B. IBM MQ Explorer, und einen Kanal für eine solche spezifische Nutzung konfiguriert haben, möchten Sie vielleicht sicherstellen, dass der Kanal nur von bestimmten Client-Computern verwendet werden kann. Über einen Kanalauthentifizierungssatz können Sie sicherstellen, dass der Kanal nur von bestimmten IP-Adressen genutzt werden kann.

Wenn Sie nur mit einigen Beispielanwendungen, die als Clients ausgeführt werden, gestartet werden, lesen Sie die Informationen im Abschnitt Musterprogramme vorbereiten und ausführen , um ein Beispiel für die sichere Konfiguration des Warteschlangenmanagers unter Verwendung von Kanalauthentifizierungsdatensätzen zu erhalten.

Die Kanalauthentifizierungsdatensätze zum Steuern eingehender Kanäle werden mit dem MQSC-Befehl **ALTER QMGR CHLAUTH(ENABLED)** abgerufen.

CHLAUTH-Regeln werden für einen MCA des Kanals angewendet, der als Antwort auf eine neue eingehende Verbindung erstellt wird. Für einen Kanal-MCA, der als Antwort auf den lokalen Start des Kanals erstellt wurde, werden keine **CHLAUTH**-Regeln angewendet.

Tabelle 6. Dabei werden CHLAUTH-Regeln für verschiedene Kanalpaare angewendet

Kanaltyp	**MCA, auf dem CHLAUTH-Regeln angewendet werden**
SDR-RCVR	RCVR
RQSTR-SVR (gestartet auf SVR)	RQSTR
RQSTR-SVR (gestartet auf RQSTR)	SVR
RQSTR-SDR (Gestartet bei SDR)	RQSTR
RQSTR-SDR (Gestartet bei RQSTR)	SDR für die Anfangsverbindung. RQSTR für die Callback-Verbindung.

Kanalauthentifizierungsdatensätze können für die folgenden Funktionen erstellt werden:

- Blockieren von Verbindungen von einer bestimmten IP-Adresse
- Blockieren von Verbindungen von bestimmten Benutzer-IDs
- Festlegen eines MCAUSER-Werts zur Verwendung für Kanäle, die von einer bestimmten IP-Adresse aus Verbindungen herstellen
- Festlegen eines MCAUSER-Werts zur Verwendung für Kanäle, die eine bestimmte Benutzer-ID bestätigen
- Festlegen eines MCAUSER-Werts zur Verwendung für Kanäle mit einem bestimmten SSL oder TLS Distinguished Name (DN)
- Festlegen eines MCAUSER-Werts zur Verwendung für Kanäle, die von einem bestimmten Warteschlangenmanager aus Verbindungen herstellen
- Blockieren von Verbindungen, die behaupten, von einem bestimmten Warteschlangenmanager zu stammen - ausgenommen, die Verbindung stammt von einer bestimmten IP-Adresse
- Blockieren von Verbindungen, die ein bestimmtes SSL- oder TLS-Zertifikat vorweisen - ausgenommen, die Verbindung stammt von einer bestimmten IP-Adresse

Diese Verwendungsmöglichkeiten werden im Folgenden näher erläutert.

Sie erstellen, ändern oder entfernen Kanalauthentifizierungsdatensätze mit dem MQSC-Befehl **SET CHLAUTH** oder dem PCF-Befehl **Set Channel Authentication Record**.

Anmerkung: Eine große Anzahl von Kanalauthentifizierungsdatensätzen kann sich negativ auf die Leistung eines Warteschlangenmanagers auswirken.

IP-Adressen blockieren

In der Regel hat die Firewall die Aufgabe, den Zugriff von bestimmten IP-Adressen aus zu verhindern. Es kann jedoch Fälle geben, in denen es zu Verbindungsversuchen von einer IP-Adresse aus kommt, die eigentlich keinen Zugriff auf Ihr IBM MQ-System haben sollte, und die Adresse vorübergehend blockiert werden muss, bevor die Firewall aktualisiert werden kann. Diese Verbindungsversuche gehen unter Umständen nicht von IBM MQ-Kanälen aus, sondern von anderen Socketanwendungen, für die fälschlicherweise Ihr IBM MQ-Empfangsprogramm als Ziel konfiguriert wurde. IP-Adressen werden mit einem Kanalauthentifizierungsdatensatz des Typs BLOCKADDR blockiert. Dabei können Sie eine oder mehrere einzelne Adressen, Adressenbereiche oder Adressengruppen unter Verwendung von Platzhaltern angeben.

Wird eine eingehende Verbindung zurückgewiesen, weil die IP-Adresse auf diese Weise blockiert ist, wird, sofern Kanalereignisse aktiviert sind und der Warteschlangenmanager aktiv ist, die Ereignisnachricht MQRC_CHANNEL_BLOCKED mit Ursachencode MQRQ_CHANNEL_BLOCKED_ADDRESS ausgegeben. Außerdem wird die Verbindung vor Rückgabe des Fehlers 30 Sekunden lang offen gehalten. Dadurch wird

sichergestellt, dass das Empfangsprogramm nicht durch wiederholte Verbindungsversuche, die ebenfalls blockiert werden, überflutet wird.

Wenn Sie IP-Adressen nur auf bestimmten Kanälen blockieren möchten oder der Fehler unverzögert ausgegeben werden soll, konfigurieren Sie einen Kanalauthentifizierungssatz des Typs ADDRESSMAP mit dem Parameter USERSRC(NOACCESS).

Immer wenn eine eingehende Verbindung aus diesem Grund zurückgewiesen wird, wird die Ereignisnachricht MQRQ_CHANNEL_BLOCKED_NOACCESS mit Ursachencode MQRQ_CHANNEL_BLOCKED_NOACCESS ausgegeben, sofern Kanalereignisse aktiviert sind und der Warteschlangenmanager aktiv ist.

Ein Beispiel finden Sie unter „Blockieren bestimmter IP-Adressen" auf Seite 410.

Benutzer-IDs blockieren

Um zu verhindern, dass bestimmte Benutzer-IDs über einen Clientkanal eine Verbindung herstellen, können Sie einen Kanalauthentifizierungssatz des Typs BLOCKUSER konfigurieren. Dieser Kanalauthentifizierungsdatensatz gilt nur für Clientkanäle, nicht für Nachrichtenkanäle. Sie können eine oder mehrere einzelne Benutzer-IDs angeben, die blockiert werden sollen; Platzhalterzeichen sind jedoch nicht zulässig.

Bei jeder eingehenden Verbindung, die aus diesem Grund zurückgewiesen wird, wird eine MQRC_CHANNEL_BLOCKED-Ereignisnachricht mit dem Qualifikationsmerkmal MQRQ_CHANNEL_BLOCKED_USERID für die Ursache ausgegeben. Voraussetzung ist, dass Kanalereignisse aktiviert sind.

Ein Beispiel finden Sie unter „Blockieren bestimmter Benutzer-IDs" auf Seite 412.

Sie können auch für bestimmte Benutzer-IDs alle Zugriffe auf bestimmte Kanäle blockieren, indem Sie einen Kanalauthentifizierungsdatensatz des Typs USERMAP unter Angabe des Parameters USERSRC(NOACCESS) setzen.

Immer wenn eine eingehende Verbindung aus diesem Grund zurückgewiesen wird, wird die Ereignisnachricht MQRQ_CHANNEL_BLOCKED_NOACCESS mit Ursachencode MQRQ_CHANNEL_BLOCKED_NOACCESS ausgegeben, sofern Kanalereignisse aktiviert sind und der Warteschlangenmanager aktiv ist.

Ein Beispiel finden Sie unter „Blockierung des Zugriffs für eine Clientbenutzer-ID" auf Seite 415.

Warteschlangenmanagernamen blockieren

Wenn festgelegt werden soll, dass der Zugriff aller Kanäle blockiert werden soll, die eine Verbindung von einem bestimmten Warteschlangenmanager aus herstellen, müssen Sie einen Kanalauthentifizierungsdatensatz des Typs QMGRMAP unter Angabe des Parameters USERSRC(NOACCESS) setzen. Sie können einen einzigen Warteschlangenmanagernamen oder eine Gruppe von Warteschlangenmanagern unter Angabe von Platzhalterzeichen angeben. Es gibt keine entsprechende BLOCKUSER-Funktion für die Blockierung von Zugriffen von Warteschlangenmanagern aus.

Immer wenn eine eingehende Verbindung aus diesem Grund zurückgewiesen wird, wird die Ereignisnachricht MQRQ_CHANNEL_BLOCKED_NOACCESS mit Ursachencode MQRQ_CHANNEL_BLOCKED_NOACCESS ausgegeben, sofern Kanalereignisse aktiviert sind und der Warteschlangenmanager aktiv ist.

Ein Beispiel finden Sie unter „Zugriff von einem fernen WS-Manager aus sperren" auf Seite 415.

SSL- oder TLS-DNs blockieren

Soll Benutzern der Zugriff verwehrt werden, die ein persönliches SSL- oder TLS-Zertifikat übergeben, das einen bestimmten definierten Namen (DN; Distinguished Name) enthält, müssen Sie einen Kanalauthentifizierungsdatensatz des Typs SSLPEERMAP unter Angabe des Parameters USERSRC(NOACCESS) setzen. Sie können einen einzelnen definierten Namen oder ein Muster mit Platzhalterzeichen angeben. Es gibt keine entsprechende BLOCKUSER-Funktion für die Blockierung von Zugriffen für definierte Namen.

Immer wenn eine eingehende Verbindung aus diesem Grund zurückgewiesen wird, wird die Ereignisnachricht MQRQ_CHANNEL_BLOCKED_NOACCESS mit Ursachencode MQRQ_CHANNEL_BLOCKED_NOACCESS ausgegeben, sofern Kanalereignisse aktiviert sind und der Warteschlangenmanager aktiv ist.

Ein Beispiel finden Sie unter „Blockungszugriff für einen definierten SSL-oder TLS-Namen" auf Seite 416.

IP-Adressen zu verwendenden Benutzer-IDs zuordnen

Wenn festgelegt werden soll, dass alle Kanäle, die eine Verbindung von einer angegebenen IP-Adresse aus herstellen, einen bestimmten MCAUSER-Wert verwenden sollen, müssen Sie einen Kanalauthentifizierungsdatensatz des Typs ADDRESSMAP setzen. Sie können eine einzelne Adresse, einen Adressenbereich oder eine Adressengruppe unter Angabe von Platzhalterzeichen angeben.

Wenn Sie eine Portweiterleitungsfunktion, Sitzungsabbruch in der DMZ (Demilitarized Zone) oder eine andere Konfiguration verwenden, bei der die dem Warteschlangenmanager präsentierte IP-Adresse geändert wird, ist die Zuordnung von IP-Adressen unter Umständen nicht geeignet für Sie.

Ein Beispiel finden Sie unter „Zuordnen einer IP-Adresse zu einer MCAUSER-Benutzer-ID" auf Seite 417.

Warteschlangenmanagernamen zu verwendenden Benutzer-IDs zuordnen

Wenn festgelegt werden soll, dass alle Kanäle, die eine Verbindung von einem angegebenen Warteschlangenmanager aus herstellen, einen bestimmten MCAUSER-Wert verwenden sollen, müssen Sie einen Kanalauthentifizierungsdatensatz des Typs QMGRMAP setzen. Sie können einen einzigen Warteschlangenmanagernamen oder eine Gruppe von Warteschlangenmanagern unter Angabe von Platzhalterzeichen angeben.

Ein Beispiel finden Sie unter „Zuordnung eines fernen Warteschlangenmanagers zu einer MCAUSER-Benutzer-ID" auf Seite 413.

Benutzer-IDs, auf die ein Client besteht, zu verwendenden Benutzer-IDs zuordnen

Wenn Sie angeben möchten, dass bei einer Verbindung von einem IBM MQ-Client unter Verwendung einer bestimmten Benutzer-ID ein anderer, vorgegebener MCAUSER-Wert verwendet werden soll, müssen Sie einen Kanalauthentifizierungsdatensatz des Typs USERMAP festlegen. Bei der Zuordnung von Benutzer-IDs sind Platzhalterzeichen nicht zulässig.

Ein Beispiel finden Sie in „Zuordnung einer Clientbenutzer-ID zu einer MCAUSER-Benutzer-ID" auf Seite 414 .

SSL- oder TLS-DNs zu verwendenden Benutzer-IDs zuordnen

Wenn festgelegt werden soll, dass alle Benutzer, die ein persönliches SSL/TLS-Zertifikat mit einem angegebenen definierten Namen (DN) übergeben, einen bestimmten MCAUSER-Wert verwenden sollen, müssen Sie einen Kanalauthentifizierungsdatensatz des Typs SSLPEERMAP setzen. Sie können einen einzelnen definierten Namen oder ein Muster mit Platzhalterzeichen angeben.

Ein Beispiel finden Sie unter „Zuordnen eines SSL-oder TLS-definierten Namens zu einer MCAUSER-Benutzer-ID" auf Seite 414.

Warteschlangenmanager, Clients oder definierte SSL-/TLS-Namen abhängig von IP-Adresse zuordnen

In einigen Fällen kann es geschehen, dass Dritte den Namen eines Warteschlangenmanagers vortäuschen (Spoofing). Ebenso kann es passieren, dass ein SSL- oder TLS-Zertifikat oder eine Schlüsseldatei gestohlen oder wiederverwendet wird. Um sich gegen diese Bedrohungen zu schützen, können Sie festlegen, dass eine Verbindung, die von einem bestimmten Warteschlangenmanager oder Client hergestellt wird, oder eine Verbindung, die einen bestimmten definierten Namen (DN) verwendet, von einer bestimmten IP-Adresse ausgehen muss. Konfigurieren Sie einen Kanalauthentifizierungssatz des Typs USERMAP, QMGRMAP oder SSLPEERMAP und geben Sie mit dem Parameter ADDRESS die zulässige IP-Adresse oder das zulässige IP-Adressmuster an.

Ein Beispiel finden Sie in „Zuordnung eines fernen Warteschlangenmanagers zu einer MCAUSER-Benutzer-ID" auf Seite 413 .

Interaktion zwischen Kanalauthentifizierungsdatensätzen

Es besteht die Möglichkeit, dass für einen Kanal, über den ein Verbindungsversuch erfolgt, mehrere Kanalauthentifizierungssätze zutreffen, was zu widersprüchlichen Auswirkungen führen kann. So kann es beispielsweise sein, dass ein Kanal eine Benutzer-ID bestätigt, die von einem Kanalauthentifizierungsda-

tensatz des Typs BLOCKUSER blockiert wird, die jedoch über ein SSL- oder TLS-Zertifikat verfügt, das mit einem Kanalauthentifizierungsdatensatz des Typs SSLPEERMAP übereinstimmt, mit dem eine andere Benutzer-ID gesetzt wird. Wenn in Kanalauthentifizierungsdatensätzen außerdem Platzhalterzeichen verwendet werden, stimmt eine IP-Adresse, ein Warteschlangenmanagername oder ein SSL- oder TLS-DN unter Umständen mit mehreren Mustern überein. Beispiel: Die IP-Adresse 192.0.2.6 entspricht den Mustern 192.0.2.0-24, 192.0.2. * und 192.0.*.6. Die entsprechende Maßnahme wird wie folgt festgelegt.

- Der verwendete Kanalauthentifizierungsdatensatz wird wie folgt ausgewählt:
  - Ein Kanalauthentifizierungsdatensatz, der genau mit dem Kanalnamen übereinstimmt, hat Priorität vor einem Kanalauthentifizierungsdatensatz, der mit dem Kanalnamen unter Verwendung eines Platzhalterzeichens übereinstimmt.
  - Ein Kanalauthentifizierungsdatensatz mit einem SSL- oder TLS-DN hat Priorität vor einem Kanalauthentifizierungsdatensatz, der eine Benutzer-ID, einen Warteschlangenmanagernamen oder eine IP-Adresse verwendet.
  - Ein Kanalauthentifizierungsdatensatz mit einer Benutzer-ID oder einem Warteschlangenmanagernamen hat Priorität vor einem Kanalauthentifizierungsdatensatz mit einer IP-Adresse.
- Wird ein entsprechender Kanalauthentifizierungsdatensatz gefunden, in dem ein MCAUSER-Wert angegeben ist, wird dieser MCAUSER-Wert dem Kanal zugeordnet.
- Wird ein entsprechender Kanalauthentifizierungsdatensatz gefunden, in dem angegeben ist, dass der Kanal keinen Zugriff hat, wird dem Kanal der MCAUSER-Wert *NOACCESS zugeordnet. Dieser Wert kann später von einem Sicherheitsexitprogramm geändert werden.
- Wird kein entsprechender Kanalauthentifizierungsdatensatz gefunden oder wurde einer gefunden, in dem angegeben ist, dass die Benutzer-ID des Kanals verwendet werden soll, wird das MCAUSER-Feld überprüft.
  - Ist das MCAUSER-Feld leer, wird dem Kanal die Client-Benutzer-ID zugeordnet.
  - Ist das MCAUSER-Feld nicht leer, wird dem Kanal der MCAUSER-Wert zugeordnet.
- Ein Sicherheitsexitprogramm wird ausgeführt. Dieses Exitprogramm setzt unter Umständen die Kanalbenutzer-ID oder legt fest, dass der Zugriff blockiert werden soll.
- Wird die Verbindung blockiert oder ist MCAUSER auf *NOACCESS gesetzt, wird der Kanal beendet.
- Wird die Verbindung außer für einen Clientkanal für keinen Kanal blockiert, wird die in den vorherigen Schritten ermittelte Kanalbenutzer-ID mit einer Liste blockierter Benutzer verglichen.
  - Ist die Benutzer-ID in der Liste mit den blockierten Benutzern enthalten, wird der Kanal beendet.
  - Ist die Benutzer-ID nicht in der Liste mit den blockierten Benutzern enthalten, wird der Kanal ausgeführt.

Wenn mehrere Kanalauthentifizierungsdatensätze mit einem Kanalnamen, einer IP-Adresse, einem Hostnamen, einem Warteschlangenmanagernamen oder einem SSL- oder TLS-DN übereinstimmen, wird die genaueste Übereinstimmung verwendet. Dabei wird wie folgt vorgegangen:

- Die größtmögliche Übereinstimmung ist ein Name ohne Platzhalterzeichen; Beispiel:
  - Ein Kanalname wie beispielsweise A.B.C
  - Eine IP-Adresse wie beispielsweise 192.0.2.6
  - Hostname von `hursley.ibm.com`
  - Ein Warteschlangenmanagername wie beispielsweise 192.0.2.6
- Die allgemeinste Übereinstimmung ist ein einzelner Stern (*), der zum Beispiel Folgendes abdeckt:
  - Alle Kanalnamen
  - Alle IP-Adressen
  - Alle Hostnamen
  - Alle Warteschlangenmanagernamen
- Ein Muster mit einem Stern am Anfang einer Zeichenfolge ist allgemeiner als ein definierter Wert am Anfang einer Zeichenfolge:

  - Bei Kanälen ist *.B.C allgemeiner als A.*
  - Bei IP-Adressen ist *.0.2.6 allgemeiner als 192.*
  - Bei Hostnamen ist `*.ibm.com` allgemeiner als `hursley.*`.
  - Bei Warteschlangenmanagernamen ist *QUEUEMANAGER allgemeiner als QUEUEMANAGER*
- Ein Muster mit einem Stern an einer bestimmten Stelle in einer Zeichenfolge ist allgemeiner als ein definierter Wert an derselben Stelle in einer Zeichenfolge (gilt entsprechend für alle nachfolgenden Stellen in einer Zeichenfolge:
  - Bei Kanälen ist A.*.C allgemeiner als A.B.*
  - Bei IP-Adressen ist 192.*.2.6 allgemeiner als 192.0.*.
  - Bei Hostnamen ist `hursley.*.com` allgemeiner als `hursley.ibm.*`.
  - Bei Warteschlangenmanagernamen ist Q*MANAGER allgemeiner als QUEUE*
- Enthalten zwei oder mehr Muster einen Stern an einer bestimmten Stelle innerhalb einer Zeichenfolge, ist das Muster mit der geringeren Anzahl an Namensbestandteilen hinter dem Stern das allgemeinere Muster:
  - Bei Kanälen ist A.* allgemeiner als A.*.C.
  - Bei IP-Adressen ist 192 .* allgemeiner als 192.*.2.*.
  - Bei Hostnamen ist `hurlsey.*` allgemeiner als `hursley.*.com`.
  - Bei Warteschlangenmanagernamen ist Q* allgemeiner als Q*MGR
- Zusätzlich gilt für eine IP-Adresse:
  - Ein mit Bindestrich (-) angegebener Bereich ist spezifischer als die Angabe eines Sterns; daher ist 192.0.2.0-24 spezifischer als 192.0.2.*.
  - Ein Bereich, bei dem es sich um die Teilmenge eines Bereichs handelt, ist spezifischer als der übergeordnete Bereich. Daher ist 192.0.2.5-15 spezifischer als 192.0.2.0-24.
  - Sich überlappende Bereiche sind nicht zulässig. So dürfen keine Kanalauthentifizierungsdatensätze für 192.0.2.0-15 und 192.0.2.10-20 definiert werden.
  - Ein Muster darf nicht weniger als die erforderliche Anzahl an Adresssegmenten enthalten, es sei denn, das letzte Zeichen ist ein einzelner Stern. Beispiel: 192.0.2 ist ungültig, aber 192.0.2.* ist gültig.
  - Ein abschließender Stern muss durch das geeignete Trennzeichen (ein Punkt (.) für IPv4, ein Doppelpunkt (:) für IPv6) vom Rest der Adresse getrennt werden. So ist 192.0* beispielsweise ungültig, da der Stern nicht getrennt ist und daher kein eigenes Segment darstellt.
  - Ein Muster kann weitere Sterne enthalten, sofern kein Stern direkt neben dem abschließenden Stern steht. Beispiel: 192.*.2.* ist gültig, aber 192.0.*.* ist ungültig.
  - Ein IPv6-Adressmuster darf keinen doppelten Doppelpunkt und keinen abschließenden Stern enthalten, da die Adresse dadurch mehrdeutig wäre. So kann 2001::* beispielsweise 2001:0000:*, 2001:0000:0000:* usw. darstellen.
- Bei einem SSL- oder TLS-DN gilt für die DN-Unterzeichenfolgen die folgende Reihenfolge:

Tabelle 7. Rangordnung von Unterzeichenfolgen

Reihenfolge	**DN-Unterzeichenfolge**	**Name**
1	SERIALNUMBER=	Seriennummer des Zertifikats
2	MAIL=	E-Mail-Adresse
3	E =	E-Mail-Adresse (wird nicht weiter unterstützt; MAIL wird verwendet)
4	UID=, USERID=	Benutzer-ID

Tabelle 7. Rangordnung von Unterzeichenfolgen (Forts.)		
Reihenfolge	**DN-Unterzeichenfolge**	**Name**
5	CN=	Allgemeiner Name
6	T =	Titel
7	OU=	Organisationseinheit
8	DC=	Domänenkomponente
9	O=	Organization
10	STREET=	Straße / Erste Adresszeile
11	L=	Ort
12	ST=, SP=, S=	Bundesland
13	PZ =	Postleitzahl
14	C =	Land
15	UNSTRUCTUREDNAME=	Hostname
16	UNSTRUCTUREDADDRESS=	IP-Adresse
17	DNQ=	Qualifikationsmerkmal für den definierten Namen

Wird beispielsweise ein SSL- oder TLS-Zertifikat mit einem DN übergeben, der die Unterzeichenfolgen O=IBM und C=UK enthält, gibt IBM MQ einem Kanalauthentifizierungsdatensatz für O=IBM den Vorzug vor dem für C=UK (wenn beide vorhanden sind).

Ein definierter Name kann mehrere Organisationseinheiten (OUs) enthalten, die in hierarchischer Reihenfolge (zuerst die großen Organisationseinheiten) angegeben werden müssen. Wenn zwei definierte Namen bis auf ihre OU-Werte identisch sind, wird der spezifischere definierte Name wie folgt bestimmt:

1. Unterscheiden sich die DNs in der Anzahl der OU-Attribute, ist der DN mit den meisten OU-Werten der spezifischere. Dies liegt daran, dass der DN mit der größeren Anzahl an Organisationseinheiten eine ausführlichere Beschreibung des DN darstellt und daher mehr Übereinstimmungskriterien bereitstellt. Selbst wenn die Organisationseinheit der höchsten Ebene ein Platzhalterzeichen ist (OU=*), wird der definierte Name mit mehr OUs weiterhin als insgesamt spezifischer betrachtet.
2. Verfügen beide DNs über dieselbe Anzahl an OU-Attributen, werden die entsprechenden OU-Paare wie folgt von links nach rechts miteinander verglichen; dabei ist das OU-Attribut ganz links die Organisationseinheit der höchsten Ebene und daher am wenigsten spezifisch:
   a. Ein OU-Attribut ohne Platzhalterzeichen ist das spezifischste, da es nur mit genau einer Zeichenfolge übereinstimmen kann.
   b. Auf Platz zwei in der Rangfolge liegt ein OU-Attribut mit einem einzigen Platzhalterzeichen am Anfang (z. B. OU=*ABC) oder am Ende (z. B. OU=ABC*).
   c. Auf Platz drei in der Rangfolge liegt ein OU-Attribut mit zwei Platzhalterzeichen (z. B. OU=*ABC*).
   d. Am wenigsten spezifisch ist ein OU-Attribut, das nur aus einem einzige Stern (OU=*) besteht.
3. Stellt sich beim Zeichenfolgevergleich heraus, dass zwei Attribute gleich spezifisch oder unspezifisch sind, wird der längeren Attributzeichenfolge als der spezifischeren der Vorzug gegeben.
4. Wird beim Zeichenfolgevergleich festgestellt, dass zwei Attributwerte gleich spezifisch oder unspezifisch sind und darüber hinaus dieselbe Länge haben, wird das Ergebnis durch einen Zeichenfolgevergleich (bei dem die Groß-/Kleinschreibung nicht beachtet wird) des DN-Teils ermittelt, wobei alle Platzhalter ausgeschlossen werden.

Wenn zwei definierte Namen bis auf ihre DC-Werte identisch sind, gelten dieselben Abgleichsregeln wie für OU-Werte, außer dass in DC-Werten das DC-Attribut ganz links der niedrigsten Ebene (größte Spezifikation) entspricht und sich die Vergleichsreihenfolge entsprechend ändert.

Kanalauthentifizierungsdatensätze anzeigen

Kanalauthentifizierungsdatensätze können mit dem MQSC-Befehl **DISPLAY CHLAUTH** oder dem PCF-Befehl **Inquire Channel Authentication Records** angezeigt werden. Dabei können Sie angeben, ob alle Datensätze zurückgegeben werden sollen, die dem übergebenen Kanalnamen entsprechen, oder ob eine genaue Übereinstimmung zurückgegeben werden soll. Die genaue Übereinstimmung zeigt, welcher Kanalauthentifizierungsdatensatz verwendet wird, wenn ein Kanal eine Verbindung von einer bestimmten IP-Adresse oder einem bestimmten Warteschlangenmanager aus oder aber unter Verwendung einer bestimmten Benutzer-ID und (optional) eines persönlichen SSL/TLS-Zertifikats mit einer bestimmten DN herstellt.

Zugehörige Konzepte

„Sicherheit für fernes Messaging” auf Seite 106
Dieser Abschnitt befasst sich mit Aspekten der Sicherheit im fernen Messaging.

Interaction von CHLAUTH und CONNAUTH

Interaktionsweise von Kanalauthentifizierungsdatensätzen (CHLAUTH) und Verbindungsauthentifizierung (CONNAUTH) in IBM MQ im Falle eines einzelnen Datenaustauschs in einem Kanal.

Verschiedene Typen von Bindungen

IBM MQ unterstützt zwei Methoden, mit denen eine Anwendung eine Verbindung herstellen kann:

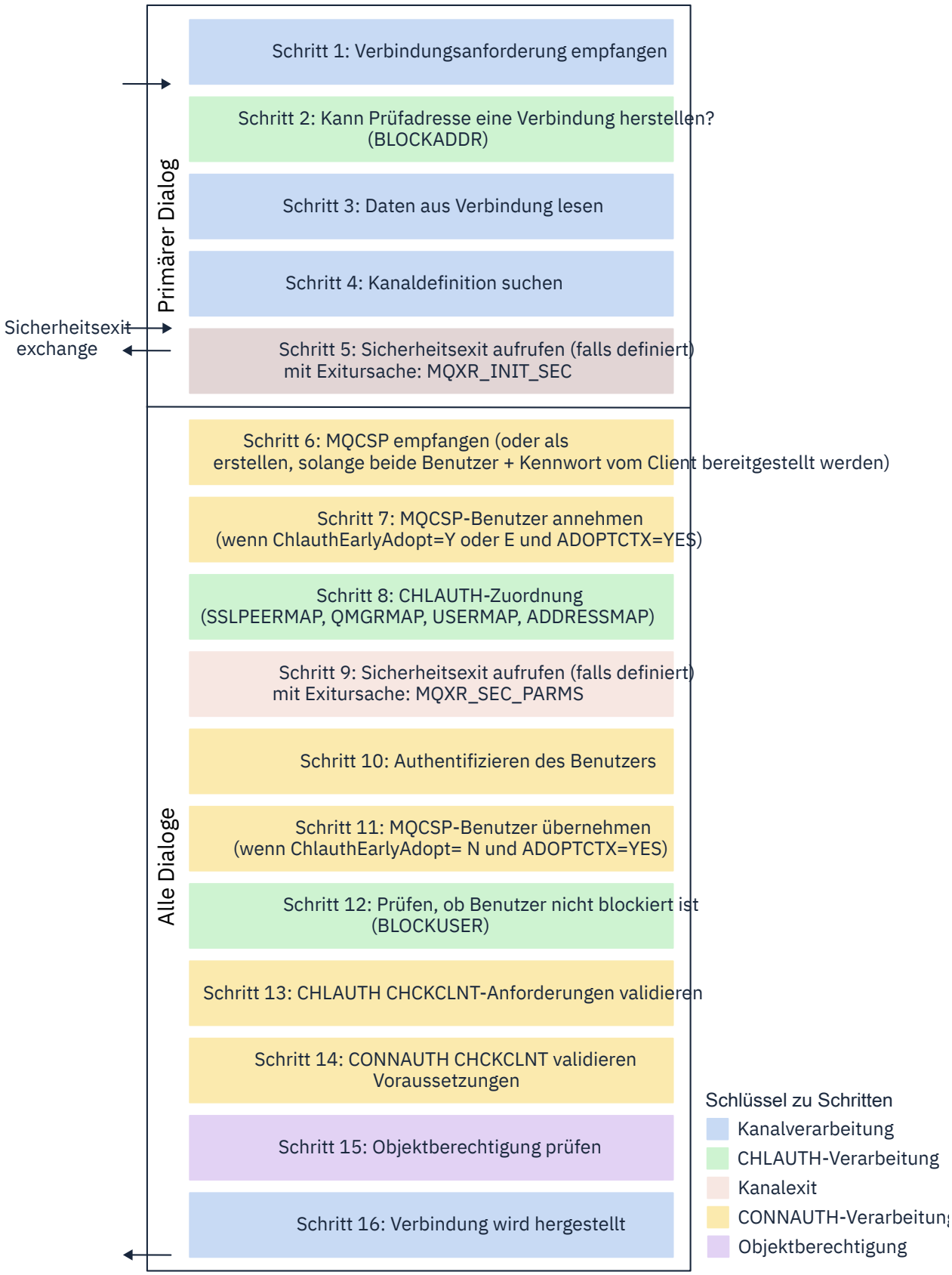
Lokale Bindungen
Gilt, wenn sich die Anwendung und der Warteschlangenmanager in demselben Betriebsimage befinden. CHLAUTH ist für diese Art von Anwendungsverbindung nicht relevant.

Clientbindungen
Gilt, wenn die Anwendung und der WS-Manager das Netz für die Kommunikation verwenden. Die Anwendung und der WS-Manager können auf derselben Maschine ausgeführt werden, oder sie können sich auf verschiedenen Maschinen befinden. In IBM MQwird eine Clientverbindung in Form eines Serververbindungskanals (SVRCONN) behandelt, und in dieser Situation ist sowohl CONNAUTH als auch CHLAUTH anwendbar.

Verbindliche Schritte des empfangenden Endes eines Kanals

Wenn eine Anwendung eine Verbindung zu einem WS-Manager herstellt, wird eine beträchtliche Anzahl von Überprüfungen ausgeführt, um sicherzustellen, dass beide Enden des Kanals die vom anderen Ende unterstützte Anzahl von Endstellen verstehen. Das empfangende Ende des Kanals führt eine zusätzliche Prüfung durch CHLAUTH und CONNAUTH durch, um sicherzustellen, dass der Client eine Verbindung herstellen darf, und dieser Prozess kann auch einen Sicherheitsexit enthalten, da dies Auswirkungen auf das Ergebnis haben kann. Diese Kanalverbindungs-Phase wird auch als *Bindungsphase* bezeichnet.

Im folgenden Diagramm werden die Schritte aufgelistet, die ein SVRCONN-Kanal durchläuft, wenn das Serverende (auf dem Queue Manager) gestartet wird:



Schritt 1: Verbindungsanforderung empfangen
Der Kanalinitiator oder Listener empfängt eine Verbindungsanforderung von einem Ort im Netz.

Schritt 2: Ist die Adresse berechtigt, eine Verbindung herzustellen?
Bevor Daten gelesen werden, überprüft IBM MQ die IP-Adresse des Partners anhand der CHLAUTH-Regeln, um festzustellen, ob die Adresse in der *BLOCKADDR*-Regel enthalten ist. Wenn die Adresse nicht gefunden wird und daher nicht blockiert wird, wird der Nachrichtenfluss mit dem nächsten Schritt fortgesetzt.

Schritt 3: Daten aus dem Kanal lesen
IBM MQ liest die Daten jetzt in einen Puffer und beginnt, die gesendeten Informationen zu verarbeiten.

Schritt 4: Suchen Sie die Kanaldefinition.
Im ersten Datenfluss sendet IBM MQ unter anderem den Namen des Kanals, den das sendende Ende versucht zu starten. Der empfangende Warteschlangenmanager kann dann die Kanaldefinition suchen, die über alle Einstellungen verfügt, die für den Kanal angegeben sind.

Schritt 5: Sicherheitsexit anrufen (falls definiert)
Wenn für den Kanal ein Sicherheitsexit (SCYEXIT) definiert ist, wird dieser aufgerufen, wobei der Exit-Grund (MQCXP.**ExitReason**) auf MQXR_INIT_SEC gesetzt wird.

Schritt 6: MQCSP empfangen
Falls erforderlich, bauen Sie eine aus, solange die Benutzer-ID und das Kennwort vom Client bereitgestellt werden.

Wenn es sich beim Client um eine Java- oder JMS-Anwendung handelt, die im Kompatibilitätsmodus ausgeführt wird, übergibt der Client keine MQCSP-Struktur an den Warteschlangenmanager. Wenn in der Anwendung eine Benutzer-ID und ein Kennwort angegeben ist, wird stattdessen an dieser Stelle eine MQCSP-Struktur erstellt.

Schritt 7: MQCSP-Benutzer aufnehmen (wenn`ChlauthEarlyAdopt` *Y* ist und ADOPTCTX=YES)
Die vom Client bestätigte Benutzer-ID wird authentifiziert.

Wenn CONNAUTH mithilfe von LDAP einen bestätigten definierten Namen einer kurzen Benutzer-ID zuordnet, wird die Zuordnung in diesem Schritt vorgenommen.

Bei einer erfolgreichen Authentifizierung wird die Benutzer-ID vom Kanal übernommen und im Zuordnungsschritt CHLAUTH verwendet.

Anmerkung: AbIBM MQ 9.0.4 wird der **`ChlauthEarlyAdopt=`***Y*Parameter automatisch in die Stanza für Kanäle in der Datei qm.ini für neue Warteschlangenmanager hinzugefügt.

Schritt 8: CHLAUTH-Zuordnung
Der Cache CHLAUTH wird erneut geprüft, um nach den Zuordnungsregeln *SSLPEERMAP*, *USERMAP*, *QMGRMAP* und *ADDRESSMAP* zu suchen.

Die Regel, die mit dem eingehenden Kanal übereinstimmt, wird am meisten verwendet. Wenn die Regel **USERSRC***(KANAL)* oder *(MAP)*enthält, wird die Bindung des Kanals fortgesetzt.

Wenn die CHLAUTH-Regeln zu einer Regel mit **USERSRC***(NOACCESS)*ausgewertet werden, wird die Verbindung der Anwendung zum Kanal blockiert, es sei denn, die Berechtigungsnachweise werden nachfolgend mit einer gültigen Benutzer-ID und einem gültigen Kennwort in Schritt 9überschrieben.

Schritt 9: Sicherheitsexit anrufen (falls definiert)
Wenn für den Kanal ein Sicherheitsexit (SCYEXIT) definiert ist, wird dieser aufgerufen, wobei der Exit-Grund (MQCXP.**ExitReason**) auf MQXR_SEC_PARMS gesetzt wird.

Ein Zeiger auf MQCSP ist im**SecurityParms** Feld der MQCXP-Struktur vorhanden.

Die MQCSP-Struktur hat Verweise auf die Benutzer-ID (MQCSP.**CSPUserIdPtr**) und Kennwort (MQCSP.**CSPPasswordPtr**).

Die Benutzer-ID und das Kennwort können im Exit geändert werden. Im folgenden Beispiel wird gezeigt, wie ein Sicherheitsexit die Werte für Benutzer-ID und Kennwort in einem Prüfprotokoll ausgeben würde:

```
if (pMQCXP -> ExitReason == MQXR_SEC_PARMS)
```

```
{
  /* It is not a good idea for security reasons to print out the user ID */
  /* and password but the following is shown for demonstration reasons */
  printf("User ID: %.*s Password: %.*s\n",
         pMQCXP -> SecurityParms -> CSPUserIdLength,
         pMQCXP -> SecurityParms -> CSPUserIdPtr,
         pMQCXP -> SecurityParms -> CSPPasswordLength,
         pMQCXP -> SecurityParms -> CSPPasswordPtr);
```

Der Exit kann IBM MQ anweisen, den Kanal zu schließen, indem er *MQXCC_CLOSE_CHANNEL* im MQCXP zurückgibt.Feld **Exitresponse** . Andernfalls wird die Kanalverarbeitung bis zur Verbindungs-Authentifizierungsphase fortgesetzt.

Anmerkung: Wenn der bestätigte Benutzer durch den Sicherheitsexit geändert wird, werden CHLAUTH-Zuordnungsregeln nicht erneut für den neuen Benutzer angewendet.

Schritt 10: Authentifizieren des Benutzers

Die Authentifizierungsphase tritt auf, wenn CONNAUTH auf dem WS-Manager aktiviert ist.

Um dies zu überprüfen, geben Sie den MQSC-Befehl 'DISPLAY QMGR CONNAUTH' aus.

z/OS Das folgende Beispiel zeigt die Ausgabe des Befehls **DISPLAY QMGR CONNAUTH** von einem Warteschlangenmanager unter IBM MQ for z/OS.

```
 CSQM201I !MQ25 CSQMDRTC DISPLAY QMGR DETAILS
 QMNAME(MQ25)
 CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
  END QMGR DETAILS
 CSQ9022I !MQ25 CSQMDRTC ' DISPLAY QMGR' NORMAL COMPLETION
```

Multi Das folgende Beispiel zeigt die Ausgabe des Befehls**DISPLAY QMGR CONNAUTH**von einem Warteschlangenmanager unter IBM MQ for Multiplatforms.

```
     1 : DISPLAY QMGR CONNAUTH
AMQ8408: Display Queue Manager details.
   QMNAME(DEMO)
   CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
```

Der CONNAUTH-Wert ist der Name eines **AUTHINFO** IBM MQ -Objekts.

Da die Betriebssystemauthentifizierung (**AUTHTYPE**(*IDPWOS*)) auf IBM MQ for Multiplatforms und IBM MQ for z/OSgültig ist, wird in den Beispielen die Betriebssystemauthentifizierung verwendet.

z/OS Das folgende Beispiel zeigt das gelieferte Standardobjekt für **AUTHTYPE**(*IDPWOS*) von einem Warteschlangenmanager, der auf IBM MQ for z/OS ausgeführt wird.

```
 CSQM293I !MQ25 CSQMDRTC 1 AUTHINFO FOUND MATCHING REQUEST CRITERIA
 CSQM201I !MQ25 CSQMDRTC DISPLAY AUTHINFO DETAILS
 AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
 AUTHTYPE(IDPWOS)
 QSGDISP(QMGR)
 ADOPTCTX(NO)
 CHCKCLNT(NONE)
 CHCKLOCL(OPTIONAL)
 FAILDLAY(1)
 DESCR()
 ALTDATE(2018-06-04)
 ALTTIME(10.43.04)
   END AUTHINFO DETAILS
 CSQ9022I !MQ25 CSQMDRTC ' DISPLAY AUTHINFO' NORMAL COMPLETION
```

Multi Das folgende Beispiel zeigt das gelieferte Standardobjekt für **AUTHTYPE**(*IDPWOS*) von einem Warteschlangenmanager, der auf IBM MQ for Multiplatforms ausgeführt wird.

```
 1 : display authinfo(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
AMQ8566: Display authentication information details.
   AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
   AUTHTYPE(IDPWOS)                    ADOPTCTX(NO)
```

```
DESCR( )                                CHCKCLNT(REQDADM)
CHCKLOCL(OPTIONAL)                      FAILDLAY(1)
ALTDATE(2015-06-08)                     ALTTIME(16.35.16)
```

AUTHINFO TYPE (IDPWOS) hat ein Attribut mit dem Namen CHCKCLNT . Wenn der Wert in *REQUIRED* geändert wird, müssen alle Clientanwendungen eine gültige Benutzer-ID und ein gültiges Kennwort angeben.

Wenn der Benutzer in Schritt 7 authentifiziert wurde, wird er nicht erneut authentifiziert, es sei denn, der Benutzer oder das Kennwort im Feld SecurityParms der MQCXP-Struktur wurde durch einen Sicherheitsexit in Schritt 9 geändert.

Schritt 11: Kontext des MQCSP-Benutzers übernehmen (wenn `ChlauthEarlyAdopt`=*N* und ADOPTCTX=YES)

Sie können das Attribut ADOPTCTX festlegen, das steuert, ob der Kanal unter MCAUSER ausgeführt wird, oder die Benutzer-ID, die die Anwendung angegeben hat.

Wenn die in der MQCSP oder im Feld**`SecurityParms`**der MQXCP-Struktur angegebene Benutzer-ID erfolgreich authentifiziert wurde und**ADOPTCTX** *JA* lautet, wird der sich aus den Schritten 7 und 8 ergebende Benutzerkontext als der für diese Anwendung zu verwendende Kontext übernommen, es sei denn, der Benutzer oder das Kennwort im Feld**`SecurityParms`** der MQCXP-Struktur wurde durch einen Sicherheitsexit in Schritt 9 geändert.

Bei dieser bestätigten Benutzer-ID handelt es sich um die Benutzer-ID, die auf die Berechtigung zur Verwendung von IBM MQ-Ressourcen geprüft wird.

Sie haben zum Beispiel keinen MCAUSER auf dem SVRCONN-Kanal eingestellt, und Ihr Client läuft unter '`johndoe`' auf IhremLinuxRechner. Ihre Anwendung gibt den Benutzer '`fred`' im MQCSP an, so dass der Kanal mit '`johndoe`' als dem aktiven MCAUSER startet. Nach der CONNAUTH-Prüfung wird der Benutzer '`fred`' übernommen und der Kanal läuft mit '`fred`' als dem aktiven MCAUSER.

Schritt 12: Prüfen, ob der Benutzer blockiert ist (BLOCKUSER)

Wenn die **CONNAUTH** -Prüfung erfolgreich ist, wird der CHLAUTH-Cache erneut inspiziert, um zu prüfen, ob der aktive MCAUSER durch eine *BLOCKUSER* -Regel gesperrt ist. Wenn der Benutzer blockiert ist, wird der Kanal beendet.

Step13: CHLAUTH CHCKCLNT-Anforderungen validieren

Wenn die in Schritt 8 ausgewählte CHLAUTH-Regel zusätzlich den CHCKCLNT-Wert REQUIRED oder REQDADM angibt, wird geprüft, ob eine gültige CONNAUTH-Benutzer-ID angegeben wurde, um die Anforderung zu erfüllen.

- Wenn CHCKCLNT (REQUIRED) festgelegt ist, muss ein Benutzer in Schritt 7 oder 10authentifiziert worden sein. Andernfalls wird die Verbindung zurückgewiesen.
- Wenn CHCKCLNT (REQDADM) festgelegt ist, muss ein Benutzer in Schritt 7 oder 10 authentifiziert worden sein, wenn diese Verbindung als privilegiert eingestuft wird. Andernfalls wird die Verbindung zurückgewiesen.
- Wenn CHCKCLNT (AS-Warteschlangenmanager) gesetzt ist, wird dieser Schritt übersprungen.

Anmerkungen:

1. Wenn CHCKCLNT (REQUIRED) oder CHCKCLNT (REQDADM) gesetzt ist, CONNAUTH jedoch im Warteschlangenmanager nicht aktiviert ist, schlägt die Verbindung mit dem Rückkehrcode MQRC_SECURITY_ERROR (2063) aufgrund des Konflikts in der Konfiguration fehl.
2. Der Benutzer wird in diesem Schritt nicht erneut authentifiziert.

Schritt 14: CONNAUTH CHCKCLNT-Anforderungen validieren

Die Authentifizierungsphase tritt auf, wenn CONNAUTH auf dem WS-Manager aktiviert ist.

Der Wert für CONNAUTH CHCKCLNT wird geprüft, um festzustellen, welche Anforderungen für eingehende Verbindungen festgelegt sind:

- Wenn CHCKCLNT (NONE) festgelegt ist, wird dieser Schritt übersprungen.
- Wenn CHCKCLNT (OPTIONAL) festgelegt ist, wird dieser Schritt übersprungen.
- Wenn CHCKCLNT (REQUIRED) festgelegt ist, muss ein Benutzer in Schritt 7 oder 10authentifiziert worden sein. Andernfalls wird die Verbindung zurückgewiesen.

- Wenn CHCKCLNT (REQDADM) festgelegt ist, muss ein Benutzer in Schritt 7 oder 10 authentifiziert worden sein, wenn diese Verbindung als privilegiert eingestuft wird. Andernfalls wird die Verbindung zurückgewiesen.

Anmerkung: Der Benutzer wird in diesem Schritt nicht erneut authentifiziert.

Multi **Schritt 15: Objektberechtigung prüfen**
Es wird eine Prüfung vorgenommen, um sicherzustellen, dass der aktive MCAUSER-Benutzer über die entsprechende Berechtigung für eine Verbindung zum Warteschlangenmanager verfügt.

ALW Weitere Informationen finden Sie unter Objektberechtigungsmanager .

IBM i Weitere Informationen finden Sie in „Objektberechtigungsmanager unter IBM i" auf Seite 169.

Schritt 16: Die Verbindung wird abgeschlossen.
Wenn die vorhergehenden Schritte erfolgreich abgeschlossen wurden, wird die Verbindung beendet.

Zugehörige Konzepte
VERBINDUNG
Ein Warteschlangenmanager kann so konfiguriert werden, dass er eine angegebene Benutzer-ID und ein Kennwort verwendet, um zu überprüfen, ob ein Benutzer über die Berechtigung für den Zugriff auf Ressourcen verfügt.

Zugehörige Verweise
SET CHLAUTH
ALTER AUTHINFO

CHLAUTH-Zugriffsprobleme beheben
Vorschläge zum Beheben bestimmter Zugriffsprobleme bei Verwendung von Kanalauthentifizierungsdatensätzen (CHLAUTH).

Standard-CHLAUTH-Regeln

Es gibt drei Standardregeln für CHLAUTH-Verarbeitung:

- NO ACCESS für alle Kanäle von `MQ-admin*` -Benutzern
- Kein Zugriff auf alle SYSTEM.* Kanäle nach allen Benutzern
- ALLOW-Zugriff auf den Kanal SYSTEM.ADMIN.SVRCONN (Nicht- `MQ-admin` Benutzer)

Die ersten beiden Regeln blockieren den Zugriff auf alle Kanäle. Die dritte Regel ist spezifischer und hat daher Vorrang vor den anderen beiden, wenn der Kanal der Kanal SYSTEM.ADMIN.SVRCONN ist, wodurch der Zugriff auf diesen Kanal ermöglicht wird.

Allgemeine Verbindungsfehler

CHLAUTH-Regeln werden verwendet, um festzustellen, ob ein Kanal gestartet werden kann, und sie ermöglichen die Zuordnung über MCAUSER zu einer anderen Benutzer-ID. Wenn der Kanal nicht gestartet werden kann, treten häufig die folgenden Fehler auf:

- RC 2035 MQRC_NOT_AUTHORIZED
- RC 2059 MQRC_Q_MGR_NOT_AVAILABLE
- AMQ4036 Zugriff nicht zulässig
- AMQ9776: Kanal wurde von userid blockiert
- AMQ9777: Kanal wurde blockiert
- MQJE001: Es ist eine MQException aufgetreten: Beendigungscode 2, Ursache 2035
- MQJE036: Verbindungsversuch des WS-Managers zurückgewiesen

Sie sollten den Zugriff strikt sperren und dann weitere CHLAUTH-Regeln hinzufügen, um die Kanäle zu steuern, die auf Kanäle zugreifen und diese starten können. Als temporäre Kennzahl und zur Behebung der aufgezählten Fehler können Sie folgende Schritte ausführen:

- „CHLAUTH-Regeln inaktivieren” auf Seite 67
- „CHLAUTH-Regeln ändern oder entfernen” auf Seite 67

CHLAUTH-Regeln inaktivieren

Als temporäre Kennzahl können Sie die CHLAUTH-Regeln inaktivieren und auch die oben genannten Fehler beheben. Die Regeln können jederzeit erneut aktiviert werden. Wenn die Inaktivierung der CHLAUTH-Regeln das Verbindungsproblem löst, wissen Sie, dass dies die Ursache war.

Um CHLAUTH-Regeln zu inaktivieren, geben Sie den folgenden Befehl aus:

```
runmqsc: ALTER QMGR CHLAUTH (DISABLED)
```

Beachten Sie, dass Sie CHLAUTH auch auf *WARN* setzen können, wodurch der Zugriff möglich ist und das Ergebnis der Regel protokolliert wird.

CHLAUTH-Regeln ändern oder entfernen

Sie können auch die CHLAUTH-Regel oder Regeln löschen oder ändern, wodurch Ihr Problem verursacht wird.

Um eine CHLAUTH-Regel zu ändern, verwenden Sie den Befehl SET CHLAUTH mit ACTION (REPLACE). Wenn Sie beispielsweise die Standardregel ändern möchten, die keinen Zugriff auf alle Kanäle von `MQ-admin` -Benutzern zu WARN verursacht, statt blockiert zu sein, geben Sie den folgenden Befehl aus:

```
runmqsc: SET CHLAUTH (*) TYPE (BLOCKUSER) USERLIST (*MQADMIN) WARN(YES)
ACTION (REPLACE)
```

Um eine CHLAUTH-Regel zu löschen, verwenden Sie den Befehl SET CHLAUTH mit der Aktion ACTION (REMOVE). Geben Sie beispielsweise den folgenden Befehl aus, um die Standardregel zu löschen, die keinen Zugriff auf alle Kanäle durch `MQ-admin` -Benutzer verursacht:

```
runmqsc: SET CHLAUTH (*) TYPE (BLOCKUSER) USERLIST (*MQADMIN) ACTION (REMOVE)
```

Testen des Zugriffs mit MATCH (RUNCHECK)

Sie können das Ergebnis Ihrer CHLAUTH-Regeln unter Verwendung der Option MATCH (*RUNCHECK*) der CHLAUTH-Regel in runmqsc testen. Die Option **MATCH** (*RUNCHECK*) gibt den Datensatz zurück, der zur Ausführungszeit von einem bestimmten eingehenden Kanal abgeglichen wird, wenn dieser Kanal eine Verbindung zu diesem Warteschlangenmanager herstellt. Sie müssen Folgendes angeben:

- Der Kanalname
- Attribut "ADDRESS"
- SSLPEER-Attribut, nur wenn der eingehende Kanal SSL oder TLS verwendet
- QMNAME, wenn der eingehende Kanal ein WS-Manager-Kanal ist, oder
- CLNTUSER, Attribut, wenn der eingehende Kanal ein Clientkanal ist

Im folgenden Beispiel wird überprüft, welche CHLAUTH-Regel mit den Standardregeln in einem `MQ-admin` -Benutzer `johndoe` auf einen Kanal mit dem Namen `CHAN1` zugreift:

```
runmqsc: DISPLAY CHLAUTH (CHAN1) MATCH (RUNCHECK) CLNTUSER ('johndoe') ADDRESS
('192.168.1.138')

AMQ8878: Display channel authentication record details.
```

```
CHLAUTH(*) TYPE(BLOCKUSER)
USERLIST(*MQADMIN)
```

Für Benutzer `johndoe` wird der Kanal nicht ausgeführt, der Benutzer wird aufgrund der BLOCKUSER-Regel für *MQADMIN-Benutzer geblockt.

Im folgenden Beispiel wird überprüft, welche CHLAUTH-Regel mit den Standardregeln die Benutzer `alice` , die kein `MQ-admin` -Benutzer ist, auf einen Kanal mit dem Namen `CHAN1` zugreift.

```
runmqsc: DISPLAY CHLAUTH (CHAN1) MATCH (RUNCHECK) CLNTUSER ('alice') ADDRESS
('192.168.1.138')

AMQ9783: Channel will run using MCAUSER('alice').
```

Für Benutzer `alice` wird der Kanal ausgeführt, und der Kanal übergibt `alice` in als MCAUSER. MCAUSER ist die Benutzer-ID, die zum Überprüfen von IBM MQ-Objektberechtigungen verwendet wird.

Zugehörige Verweise

SET CHLAUTH

ANZEIGEN CHLAUTH

Neue CHLAUTH-Regeln für Benutzer erstellen

Einige allgemeine Szenarios für Benutzer und z. B. CHLAUTH-Regeln, um diese zu erreichen.

Dieses Thema enthält die folgenden Szenarios:

- „Zugriff für bestimmte MQ-Benutzer mit Administratorberechtigung steuern” auf Seite 68
- „Steuern des Zugriffs für einen bestimmten Benutzer und eine IBM MQ-Clientanwendung” auf Seite 69
- „Zugriff für einen bestimmten Benutzer steuern, indem der definierte Name (DN) des Zertifikats verwendet wird” auf Seite 70
- „Zuordnen eines bestimmten Benutzers zum mqm -Benutzer” auf Seite 70

Zugriff für bestimmte MQ-Benutzer mit Administratorberechtigung steuern

Richten Sie für dieses Szenario einen Serververbindungskanal ein, der exklusiv für eine administrative Perspektive verwendet werden soll, d. h. für die Verbindung von IBM MQ Explorer. Sie haben einen bestimmten Kanal für diese Verwendung und definierte IP-Adresse oder Adressen, von dem aus Verbindungen akzeptiert werden sollen, und der Zugriff für die `'mqm'` -ID blockiert wird, wenn die Verbindung nicht von einer der angegebenen IP-Adressen entfernt wird.

Erstellen Sie einen SVRCONN-Kanal für IBM MQ Explorer und `MQ-admin`-Benutzer mit dem Namen `ADMIN.CHAN`:

```
runmqsc: DEFINE CHANNEL (ADMIN.CHAN) CHLTYPE (SVRCONN) TRPTYPE (TCP)
```

Stellen Sie zum Testen sicher, dass ein Benutzer in der Gruppe `MQ-admin` definiert ist, und zwar nicht. In diesem Szenario befindet sich `mqadm` in der Gruppe `MQ-admin` , und `alice` ist nicht vorhanden.

Die Standard-CHLAUTH-Regeln sind in Position. Fügen Sie drei Regeln hinzu, um einem bestimmten Benutzer den Zugriff auf ADMIN.CHAN als `MQ-admin` von bestimmten IP-Adressen zu ermöglichen:

- Setzen Sie NOACCESS von einer beliebigen Adresse aus.
- Setzen Sie BLOCKUSER für diesen Kanal auf den Benutzer `nobody`, der den Wert *MQADMIN BLOCKUSER überschreibt.
- ALLOW-Zugriff auf Benutzer `mqadm` in einem bestimmten Teilnetz von Adressen und MAP-zu- `mqadm` -Benutzerberechtigung

```
runmqsc:
SET CHLAUTH (ADMIN.CHAN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
SET CHLAUTH('ADMIN.CHAN') TYPE(BLOCKUSER) +
DESCR('Rule to override *MQADMIN blockuser on this channel') +
```

```
USERLIST('nobody') ACTION(replace)
SET CHLAUTH('ADMIN.CHAN') TYPE(USERMAP) +
CLNTUSER('mqadm') USERSRC(MAP) MCAUSER('mqadm') +
ADDRESS('192.168.1.*') +
DESCR('Allow mqadm as mqadm on local subnet') ACTION(ADD)
```

Zu diesem Zeitpunkt kann der Benutzer `mqadm` auf den Kanal ADMIN.CHAN aus dem angegebenen IP-Adressbereich zugreifen und diese starten.

Sie können MATCH (RUNCHECK) zu einem beliebigen Zeitpunkt ausführen, um die Ergebnisse jedes dieser Befehle anzuzeigen:

```
runmqsc:
DISPLAY CHLAUTH (ADMIN.CHAN) MATCH (RUNCHECK) CLNTUSER ('mqadm') ADDRESS
('192.168.1.138')
AMQ8878: Display channel authentication record details.
CHLAUTH(ADMIN.CHAN) TYPE(USERMAP)
ADDRESS(192.168.1.*) CLNTUSER(mqadm)
MCAUSER(mqadm)

DISPLAY CHLAUTH (ADMIN.CHAN) MATCH (RUNCHECK) CLNTUSER ('alice') ADDRESS
('192.168.1.138')
AMQ8878: Display channel authentication record details.
CHLAUTH(ADMIN.CHAN) TYPE(ADDRESSMAP)
ADDRESS(*) USERSRC(NOACCESS)
```

Zu diesem Zeitpunkt dürfen nur die Benutzer mit einem CHLAUTH-Datensatz auf die Verwendung von ADMIN.CHAN. zugreifen.

Steuern des Zugriffs für einen bestimmten Benutzer und eine IBM MQ-Clientanwendung

Für dieses Szenario sind die CHLAUTH-Standardregeln ausreichend, vorausgesetzt, die IBM MQ-Berechtigung sollte für einen bestimmten Benutzer festgelegt werden, damit die richtige IBM MQ-Berechtigung (mit setmqaut) bereitgestellt wird.

In diesem Szenario werden die Berechtigungen für einen Benutzer `mqapp1` festgelegt, der kein `MQ-admin` -Benutzer ist. Erstellen Sie einen SVRCONN-Kanal (APP1.CHAN), der von einer bestimmten Anwendung und einem bestimmten Benutzer verwendet werden soll.

```
runmqsc: DEFINE CHANNEL (APP1.CHAN) CHLTYPE (SVRCONN) TRPTYPE (TCP)
```

Mit den Standard-CHLAUTH-Regeln kann der Benutzer `mqapp1` den Kanal APP1.CHAN starten.

Die Benutzer-ID, die von der IBM MQ-Clientanwendung stammt, wird für die Objektberechtigungsüberprüfung von IBM MQ verwendet. In diesem Fall wird (vorausgesetzt, der Benutzer 'mqapp1' führt die IBM MQ-Client-App aus) diese für die Objektberechtigungsüberprüfung von IBM MQ verwendet. Wenn `mqapp1` daher Zugriff auf die IBM MQ-Objekte hat, die die Anwendung benötigt, ist alles in Ordnung. Wenn nicht, erhalten Sie Berechtigungsfehler.

Sie können die Sicherheit weiter erhöhen, indem Sie bestimmte CHLAUTH-Regeln für die `mqapp1` -Benutzer-ID erstellen, aber unter den Standardregeln kann kein Mitglied der Gruppe `MQ-admin` auf diesen Kanal zugreifen.

```
runmqsc:
SET CHLAUTH (APP1.CHAN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
SET CHLAUTH('APP1.CHAN') TYPE(USERMAP) +
CLNTUSER('mqapp1') USERSRC(MAP) MCAUSER('mqapp1') +
DESCR('Allow mqapp1 as mqapp1 on local subnet') ACTION(ADD)
```

Zugriff für einen bestimmten Benutzer steuern, indem der definierte Name (DN) des Zertifikats verwendet wird

Für dieses Szenario muss der Benutzer über ein Zertifikat verfügen, das an den Warteschlangenmanager geleitet wird. Der DN wird dann mit der SSLPEER -Einstellung der CHLAUTH-Regel abgeglichen, und der SSLPEER kann Platzhalterzeichen verwenden.

Wenn eine Übereinstimmung vorhanden ist, kann der Benutzer auch einem anderen MCAUSER zugeordnet werden, um die IBM MQ-Objektberechtigungen zu überprüfen. Durch die Zuordnung des MCAUSER-Werts kann die Anzahl der Benutzer, die im IBM MQ-Objektberechtigungsmanager (OAM) verwaltet werden müssen, minimiert werden.

Sie verfügen über einen TLS-Kanal mit Zertifikaten, die Sie verwenden, und Sie benötigen Regeln für:

- Alle Benutzer für einen bestimmten Kanal blockieren
- Ermöglichen Sie nur Benutzern mit einem bestimmten SSLPEER, die den Client dieses Benutzers für den IBM MQ-OAM-Zugriff verwenden.

```
.
# block all users on any IP address.
SET CHLAUTH('SSL1.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('*')
USERSRC(NOACCESS) DESCR(''block all'') WARN(NO) ACTION(ADD)
.
# override - no MQM admin rule (allow mqm group /mqm admin users to
connect.
SET CHLAUTH('SSL1.SVRCONN') TYPE(BLOCKUSER) USERLIST('nobody')
DESCR('override no mqm admin rule') WARN(NO) ACTION(ADD)
.
# allow particular SSLPEER, use client id coming in from channel
SET CHLAUTH('SSL1.SVRCONN') TYPE(SSLPEERMAP)
SSLPEER('CN=JOHNDOE,O=IBM,C=US') USERSRC(CHANNEL) ACTION(ADD)
```

Die Clientbenutzer-ID, die auf dem Kanal eine Verbindung herstellt, wird für die IBM MQ OAM-Berechtigung von IBM MQ-Objekten verwendet. Daher muss die Benutzer-ID über die entsprechenden IBM MQ-Berechtigungen verfügen.

Sie können die Zuordnung zu einer anderen IBM MQ-Benutzer-ID durchführen. Verwenden Sie hierfür:

```
USERSRC(MAP) MCAUSER('mquser1')
```

statt USERSRC(CHANNEL).

Zuordnen eines bestimmten Benutzers zum `mqm` -Benutzer

Dies ist eine Hinzufügung oder Änderung von „Zugriff für bestimmte MQ-Benutzer mit Administratorberechtigung steuern” auf Seite 68.

Fügen Sie die folgende CHLAUTH-Regel hinzu, um bestimmte Benutzer dem `mqm` -Benutzer zuzuordnen oder einer `MQ-admin`-Benutzer-ID, für die eine IBM MQ-Objektberechtigungskonfiguration im IBM MQ-OAM vorhanden ist.

```
runmqsc:
SET CHLAUTH('ADMIN.CHAN') TYPE(USERMAP) +
CLNTUSER ('johndoe') USERSRC(MAP) MCAUSER ('mqm') +
ADDRESS('192.168.1-100.*') +
DESCR ('Allow johndoe as MQ-admin on local subnet') ACTION (ADD)
```

Dies ermöglicht und ordnet den `johndoe` -Benutzer dem `mqm` -Benutzer für den jeweiligen Kanal ADMIN.CHAN. zu.

Zugehörige Konzepte

„CHLAUTH-Zugriffsprobleme beheben” auf Seite 66
Vorschläge zum Beheben bestimmter Zugriffsprobleme bei Verwendung von Kanalauthentifizierungsdatensätzen (CHLAUTH).

„Erstellen neuer CHLAUTH-Regeln für Kanäle” auf Seite 71

Hier finden Sie einige allgemeine Szenarien für Kanäle, die Ihnen bei der Erstellung Ihrer eigenen CHLAUTH-Regeln helfen, sowie CHLAUTH-Beispielregeln, um diese auszuführen.

Zugehörige Verweise
SET CHLAUTH
ANZEIGEN CHLAUTH

Erstellen neuer CHLAUTH-Regeln für Kanäle
Hier finden Sie einige allgemeine Szenarien für Kanäle, die Ihnen bei der Erstellung Ihrer eigenen CHLAUTH-Regeln helfen, sowie CHLAUTH-Beispielregeln, um diese auszuführen.

Dieses Thema enthält die folgenden Szenarios:

- „Erlaube nur den Zugriff auf einen bestimmten Kanal aus einem bestimmten IP-Adressbereich.” auf Seite 71
- „Blockieren Sie für einen bestimmten Kanal alle Benutzer, aber ermöglichen Sie es bestimmten Benutzern, eine Verbindung herzustellen.” auf Seite 71
- „CHLAUTH für Empfänger-und Senderkanäle verwenden” auf Seite 72

Erlaube nur den Zugriff auf einen bestimmten Kanal aus einem bestimmten IP-Adressbereich.

Für dieses Szenario gilt Folgendes:

- Kein Zugriff auf den Kanal von einer beliebigen Position aus
- Zugriff von einer bestimmten IP-Adresse oder einem bestimmten Adressbereich aus zulassen

```
runmqsc:
SET CHLAUTH('APP2.CHAN') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
WARN(NO) ACTION(ADD)
SET CHLAUTH('APP2.CHAN') TYPE(ADDRESSMAP) ADDRESS('9.95.100.1-5')
USERSRC(MAP) MCAUSER('mqapp2') ACTION(ADD)
```

Auf diese Weise kann nur der Kanal APP2.CHAN gestartet werden, wenn die Verbindung aus dem angegebenen IP-spezifischen Adressbereich stammt.

Der Benutzer, der als MCAUSER eine Verbindung herstellt, ist `mqapp2` zugeordnet und erhält deshalb die IBM MQ-OAM-Berechtigung für diesen Benutzer.

Blockieren Sie für einen bestimmten Kanal alle Benutzer, aber ermöglichen Sie es bestimmten Benutzern, eine Verbindung herzustellen.

Für dieses Szenario hat der Zugriff auf den Kanal MY.SVRCONN die Standard-CHLAUTH-Regeln in der Position.

Sie müssen Folgendes hinzufügen:

```
# block all users
SET CHLAUTH('MY.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR(''block all'') WARN(NO) ACTION(ADD)

# override - no MQM admin rule
SET CHLAUTH('MY.SVRCONN') TYPE(BLOCKUSER) USERLIST('nobody') DESCR('override
no mqm admin rule') WARN(NO) ACTION(ADD)

# allow johndoe userid
SET CHLAUTH('MY.SVRCONN') TYPE(USERMAP) CLNTUSER('johndoe')
USERSRC(CHANNEL) DESCR('allow johndoe userid') ACTION(ADD)
```

Dieser erste Teil des Codes blockiert die Verbindung zu MY.SVRCONN, und der Code erlaubt nur den Kanal MY.SVRCONN, der gestartet werden soll, wenn die Verbindung von der spezifischen Benutzer-ID `johndoe` stammt.

Der Benutzer, der auf dem Kanal `johndoe` eine Verbindung herstellt, wird für die IBM MQ OAM-Berechtigung von IBM MQ-Objekten verwendet. Daher muss die Benutzer-ID über die entsprechenden IBM MQ-Berechtigungen verfügen.

Sie können die Zuordnung zu einer anderen IBM MQ-Benutzer-ID durchführen. Verwenden Sie hierfür:

```
USERSRC(MAP) MCAUSER('mquser1')
```

statt `USERSRC(CHANNEL)`.

CHLAUTH für Empfänger-und Senderkanäle verwenden

Sie können CHLAUTH-Regeln verwenden, um zusätzliche Sicherheit für Empfänger-und Senderkanäle hinzuzufügen, um den Zugriff auf den Empfängerkanal zu beschränken. Hinweis: Wenn Sie CHLAUTH-Regeln hinzufügen oder Änderungen vornehmen, gelten die aktualisierten CHLAUTH-Regeln nur beim Starten des Kanals. Wenn die Kanäle bereits aktiv sind, müssen Sie sie stoppen und erneut starten, damit die CHLAUTH-Aktualisierungen angewendet werden.

CHLAUTH-Regeln können auf jedem Kanal verwendet werden, aber es gibt einige Einschränkungen. USERMAP-Regeln gelten z. B. nur für SVRCONN-Kanäle.

Dieses Beispiel ermöglicht nur eine Verbindung von einer bestimmten IP-Adresse, um den Kanal TO.MYSVR1 zu starten:

```
# First you could lock down the channel by disallowing all
# for channel 'TO.MYSVR1', RCVR channel
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR('Back-stop rule')

# Then you could allow this channel to be started
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('192.168.1.134') USERSRC(MAP)
MCAUSER('mqapp') ACTION(ADD)
```

In diesem Beispiel wird nur die Verbindung von einem bestimmten WS-Manager aus möglich:

```
# Lock down all access:
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR('Back-stop rule')

# Then allow access from queue manager MYSVR2 and from a particular ipaddress:
SET CHLAUTH('TO.MYSVR1') TYPE(QMGRMAP) QMNAME('MYSVR2') USERSRC(MAP)
MCAUSER('mqapp') ADDRESS('192.168.1.134') ACTION(ADD)
```

Zugehörige Konzepte

„CHLAUTH-Zugriffsprobleme beheben” auf Seite 66
Vorschläge zum Beheben bestimmter Zugriffsprobleme bei Verwendung von Kanalauthentifizierungsdatensätzen (CHLAUTH).

„Neue CHLAUTH-Regeln für Benutzer erstellen” auf Seite 68
Einige allgemeine Szenarios für Benutzer und z. B. CHLAUTH-Regeln, um diese zu erreichen.

Zugehörige Verweise

SET CHLAUTH

ANZEIGEN CHLAUTH

CHLAUTH-Back-Stop-Regel erstellen

Wenn Sie über die Steuerung eingehender Verbindungen in Ihren Warteschlangenmanager nachdenken, haben Sie zwei Möglichkeiten. Sie können entweder versuchen, alle Verbindungen aufzulisten, die nicht zulässig sind, oder Sie können zunächst alle Verbindungen als nicht zulässig erklären und versuchen, alle zulässigen Verbindungen aufzulisten. Diese zweite Option wird hier beschrieben.

Informationen zu diesem Vorgang

Der Grund für die Verwendung der zweiten Option ist, dass, wenn Sie versuchen, alle Verbindungen aufzulisten, die nicht erlaubt sind, und alle nicht aufgelisteten folglich erlaubt sind, das Fehlen einer Verbindung in der Liste zur Folge hat, dass eine Verbindung, die nicht hätte zugelassen werden sollen, eine Verbindung herstellen kann, und somit eine potenzielle Sicherheitslücke verursacht.

Wenn Sie stattdessen alle Verbindungen nicht zulassen und dann diejenigen auflisten, bei denen es sich nicht um eine solche Liste handelt, handelt es sich nicht um einen Sicherheitsverstoß. Wenn für Ihr Unternehmen zusätzliche Verbindungen hinzugefügt werden müssen, handelt es sich um eine relativ einfache Task, aber es gibt keine potenzielle Sicherheitsverletzung.

Als Erstes wird eine *back-stop*-Regel erstellt, die alle Verbindungen erfasst, die nicht anderweitig von genaueren Regeln erfasst werden. Diese Regel bewirkt, dass alle fernen Verbindungen daran gehindert werden, sich an Ihren Warteschlangenmanager anhängen zu können.

Wenn Sie jedoch Bedenken gegenüber diesem Ansatz hegen, können Sie die Regel *back-stop* im Warnmodus einrichten. Weitere Informationen finden Sie im Abschnitt „2” auf Seite 73.

Vorgehensweise

1. Geben Sie den folgenden Befehl aus, um eine Back-Stop-Regel zu erstellen, die ferne Verbindungen daran hindert, sich an Ihren Warteschlangenmanager anhängen zu können:

   ```
   SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*')
   USERSRC(NOACCESS) DESCR('Back-stop rule')
   ```

   Nachdem Sie nun die Tür zu allen fernen Verbindungen geschlossen haben, können Sie beginnen, genauere Regeln festzulegen, um bestimmte Verbindungen in zu ermöglichen. For example:

   ```
   SET CHLAUTH('APPL1.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('9.20.1-3.*') USERSRC(CHANNEL)
   SET CHLAUTH('SYSTEM.ADMIN.*') TYPE(SSLPEERMAP) SSLPEER('O=IBM') USERSRC(CHANNEL)
   SET CHLAUTH('TO.QM2') TYPE(QMGRMAP) QMNAME('QM1') USERSRC(MAP) MCAUSER('QM1USER')
   SET CHLAUTH('*.SVRCONN') TYPE(USERMAP) CLNTUSER('johndoe') MCAUSER('johndoe@yourdomain')
   SET CHLAUTH('*') TYPE(SSLPEERMAP) SSLPEER('CN="John Doe"') ADDRESS('9.*') MCAUSER('johndoe')
   ```

2. Wenn Sie die Back-Stop-Regel im Warnmodus erstellen möchten, geben Sie den folgenden Befehl aus:

   ```
   SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*')
   USERSRC(NOACCESS) DESCR('Back-stop rule') WARN(YES)
   ```

   Jetzt können Sie fortfahren und alle Ihre positiven Regeln festlegen. Wenn Sie alle Regeln erstellt haben, die Sie benötigen, schalten Sie Kanalereignisse ein, indem Sie den folgenden Befehl ausgeben:

   ```
   ALTER QMGR CHLEV(EXCEPTION)
   ```

   und überwachen Sie die Warteschlange SYSTEM.ADMIN.CHANNEL.EVENT für Ereignisse, bei denen **Reason** auf MQRC_CHANNEL_BLOCKED_WARNING gesetzt wurde.

   Diese Ereignisse beschreiben die Verbindungen, die mit Ihrer Back-Stop-Regel übereinstimmen, aber da der Befehl im Warnmodus ausgeführt wird, werden die Verbindungen für den Moment nicht tatsächlich blockiert.

   Prüfen Sie jedes dieser Ereignisse und stellen Sie fest, ob für diese Verbindung eine positive Regel vorhanden sein sollte, um sie zuzulassen, oder ob sie korrekt mit der *back-stop*-Regel abgeglichen wurde. Sie können die Ausführung in diesem Modus starten und die Ereignisse überprüfen, während sie erstellt werden, bis Sie sicher sind, dass Sie alle Eingangskanäle gesehen haben und entsprechende positive Regeln für sie alle eingerichtet haben.

   An diesem Punkt können Sie die Regel *back-stop* ändern, um tatsächlich mit dem Blockieren von Verbindungen zu beginnen, deren Übereinstimmung mit dem folgenden Befehl abgeglichen wird:

   ```
   SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*')
   USERSRC(NOACCESS) DESCR('Back-stop rule') WARN(NO)
   ACTION(REPLACE)
   ```

Nicht privilegierten IBM MQ -Administrator erstellen
Wie Sie einen nicht-privilegierten IBM MQ-Administrator mit CHLAUTH erstellen.

Informationen zu diesem Vorgang

Folgende Begriffe haben im Kontext dieser Task die folgende Bedeutung:

privilegierter Benutzer
Ein Benutzer, der berechtigt ist, eine Operation auszuführen, ohne dass ihm explizit Zugriff auf diese Operation erteilt wurde. Die Benutzer in der Gruppe 'mqm' sind Beispiele für diese privilegierten Benutzer.

IBM MQ-Administrator
Bezeichnet einen Benutzer, der Verwaltungsbefehle für IBM MQabsetzen muss, z. B. **`DEFINE QLOCAL`** oder **`START CHANNEL`**.

Mit den folgenden Schritten wird ein nicht-privilegierter IBM MQ-Administrator erstellt.

Vorgehensweise

1. Erstellen Sie eine Benutzer-ID auf der Warteschlangenmanager-Maschine mit den entsprechenden Befehlen für die Plattform oder Plattformen, die Ihr Unternehmen verwendet.

   In diesem Beispiel wird der Benutzername `alice` verwendet.
2. Erteilen Sie dieser neuen Benutzerberechtigung die Berechtigung, alle Verwaltungsbefehle von IBM MQ auszugeben, indem Sie die folgende Prozedur ausführen:
   a) Starten Sie IBM MQ Explorer mit einem privilegierten Benutzer.
   b) Navigieren Sie zum *Role Based Wizard* (Rollenbasierter Assistent), indem Sie den entsprechenden Warteschlangenmanager auswählen, dann `Object Authorities` (Objektberechtigungen) und `Add Role Based Authorities` (Rollenbasierte Berechtigungen) hinzufügen.
   c) Geben Sie in der Assistentenanzeige, die angezeigt wird, die Benutzer-ID ein, die Sie im ersten Schritt erstellt haben, oder geben Sie den Gruppennamen für den Benutzer oder die Gruppe von Benutzern ein, die Sie zu nicht-privilegierten IBM MQ-Administratoren machen möchten.
   d) Richten Sie den Assistenten für vollständigen Verwaltungszugriff ein.
   e) Wenn Sie zulassen möchten, dass Ihr nicht-privilegierter IBM MQ-Administrator Nachrichten in Warteschlangen durchsuchen kann, wählen Sie dieses Kontrollkästchen ebenfalls aus.
   f) Überprüfen Sie die Befehle in der Vorschauanzeige am unteren Rand des Assistenten.

      Sie können diese Befehle schneiden und einfügen und so eigene Scripts erstellen.

      Ein Grund dafür, dies mit Ihrem eigenen Script zu tun, besteht darin, den Umfang des Zugriffs, den Sie diesem Benutzer geben, zu reduzieren. Statt Zugriff auf alle Objekte zu erteilen, möchten Sie möglicherweise Zugriff nur auf eine bestimmte Gruppe von Objekten erteilen.

      Wenn Sie **OK** im Assistenten drücken, werden die Befehle so ausgegeben, wie sie angezeigt werden.
   g) Sie müssen einige CHLAUTH-Regeln einrichten, um den Fernzugriff für diese Benutzer-ID zu ermöglichen, wenn die Voraussetzung für einen nicht-privilegierten IBM MQ-Administrator auch für den fernen Zugriff erforderlich ist.

      Davon ausgehend, dass Ihr Unternehmen die Anleitung in „CHLAUTH-Back-Stop-Regel erstellen” auf Seite 72 verwendet, müssen Sie lediglich eine Aktivierungsregel hinzufügen.

      Die Regel, die Sie erstellen, hängt vielmehr davon ab, wie Sie die Authentifizierung Ihrer fernen IBM MQ-Administratoren wählen.

      Wenn Sie eine schwache TCP/IP-Authentifizierung verwenden, können Sie eine CHLAUTH-Regel einrichten, die wie folgt aussieht:

```
  SET CHLAUTH(admin-channel-name)      TYPE(ADDRESSMAP)
ADDRESS('1.2.3.4')      USERSRC(MAP) MCAUSER('alice')
DESCR('Admin Channel - Weak TCP/IP authentication')
```

9. Wenn Sie die TLS-Authentifizierung verwenden, können Sie eine CHLAUTH-Regel einrichten, die wie folgt aussieht:

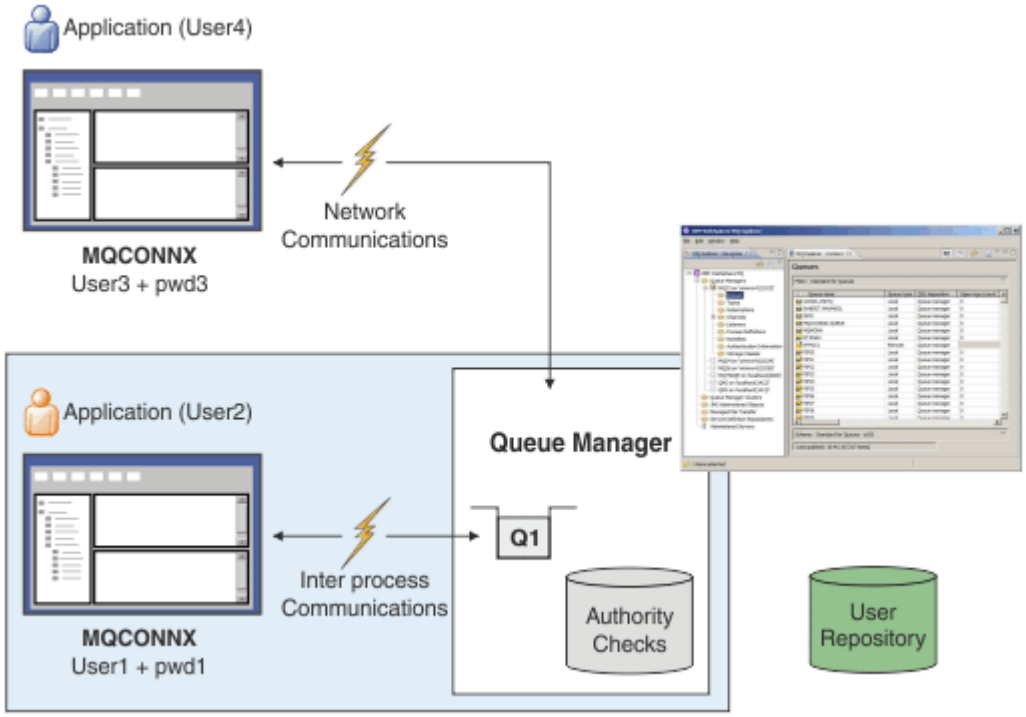
```
SET CHLAUTH(admin-channel-name)      TYPE(SSLPEERMAP)
SSLPEER('CN=Alice') ADDRESS('1.2.3.4')      USERSRC(MAP) MCAUSER('alice')
DESCR('Admin Channel - TLS authentication'
```

Wenn ein Benutzer nun eine Verbindung zum `admin-channel-name` herstellt (und mit den CHLAUTH-Regeln übereinstimmt), kann er Befehle unter der Benutzer-ID `alice` auf dem Warteschlangenmanager ausgeben, sodass ein privilegierter ferner Zugriff nicht erforderlich ist.

Verbindungsauthentifizierung

Die Verbindungsauthentifizierung kann auf verschiedene Arten erreicht werden:

- Eine Anwendung kann eine Benutzer-ID und ein Kennwort bereitstellen. Die Anwendung kann entweder ein Client sein oder lokale Bindungen verwenden.
- Ein WS-Manager kann so konfiguriert werden, dass er auf eine angegebene Benutzer-ID und ein Kennwort einwirkt.
- Es kann ein Repository verwendet werden, um zu ermitteln, ob eine Kombination aus Benutzer-ID und Kennwort gültig ist.



Im Diagramm stellen zwei Anwendungen Verbindungen zu einem Warteschlangenmanager, eine Anwendung als Client und eine Anwendung unter Verwendung von lokalen Bindungen. Anwendungen können eine Vielzahl von APIs verwenden, um eine Verbindung zum Warteschlangenmanager herzustellen. Alle haben jedoch die Möglichkeit, eine Benutzer-ID und ein Kennwort bereitzustellen. Die Benutzer-ID, unter der die Anwendung ausgeführt wird, `User2` und `User4` im Diagramm, bei denen es sich um die übliche Benutzer-ID des Betriebssystems handelt, die IBM MQangezeigt wird, können sich von der von der Anwendung bereitgestellten Benutzer-ID `User1` und `User3`unterscheiden.

Der Warteschlangenmanager empfängt Konfigurationsbefehle (im Diagramm wird IBM MQ Explorer verwendet), verwaltet das Öffnen von Ressourcen und prüft die Berechtigung für den Zugriff auf diese Ressourcen. Es gibt verschiedene Ressourcen in IBM MQ, auf die eine Anwendung möglicherweise zugreifen muss. Das Diagramm veranschaulicht das Öffnen einer Warteschlange für die Ausgabe, aber die gleichen Prinzipien gelten auch für andere Ressourcen.

Im Abschnitt Benutzerrepositorys finden Sie Details zu dem Repository, das für die Überprüfung von Benutzer-IDs und Kennwörtern verwendet wird.

Zugehörige Konzepte

„Verbindungsauthentifizierung: Konfiguration" auf Seite 76
Ein Warteschlangenmanager kann so konfiguriert werden, dass er eine angegebene Benutzer-ID und ein Kennwort verwendet, um zu überprüfen, ob ein Benutzer über die Berechtigung für den Zugriff auf Ressourcen verfügt.

„Verbindungsauthentifizierung: Anwendungsänderungen" auf Seite 80

„Verbindungsauthentifizierung: Benutzerrepositorys" auf Seite 81
Für jeden Ihrer Warteschlangenmanager können Sie verschiedene Typen von Authentifizierungsinformationsobjekten für die Authentifizierung von Benutzer-IDs und Kennwörtern auswählen.

Verbindungsauthentifizierung: Konfiguration

Ein Warteschlangenmanager kann so konfiguriert werden, dass er eine angegebene Benutzer-ID und ein Kennwort verwendet, um zu überprüfen, ob ein Benutzer über die Berechtigung für den Zugriff auf Ressourcen verfügt.

Verbindungsauthentifizierung auf einem WS-Manager aktivieren

In einem WS-Manager-Objekt kann das Attribut **CONNAUTH** auf den Namen eines Authentifizierungsinformationsobjekts (AUTHINFO) gesetzt werden. Bei diesem Objekt kann es sich um einen der beiden Typen (Attribut AUTHTYPE) handeln:

IDPWOS
: Gibt an, dass der WS-Manager das lokale Betriebssystem verwendet, um die Benutzer-ID und das Kennwort zu authentifizieren.

IDPWLDAP
: Gibt an, dass der Warteschlangenmanager einen LDAP-Server verwendet, um die Benutzer-ID und das Kennwort zu authentifizieren.

Anmerkung: Sie können keine andere Art von Authentifizierungsinformationsobjekt im Feld **CONNAUTH** verwenden.

IDPWOS und IDPWLDAP sind in einer Reihe ihrer Attribute ähnlich, die hier beschrieben werden. Andere Attribute werden später berücksichtigt.

Wenn Sie lokale Verbindungen überprüfen möchten, verwenden Sie das Attribut AUTHINFO **CHCKLOCL** (lokale Verbindungen überprüfen). Wenn Sie Clientverbindungen überprüfen möchten, verwenden Sie das Attribut AUTHINFO **CHCKCLNT** (Clientverbindungen überprüfen). Die Konfiguration muss aktualisiert werden, bevor der WS-Manager die Änderungen erkennt.

```
ALTER QMGR CONNAUTH(USE.PW)
DEFINE AUTHINFO(USE.PW) +
AUTHTYPE(IDPWOS) +
FAILDLAY(10) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED)
REFRESH SECURITY TYPE(CONNAUTH)
```

Dabei ist USE.PW in der Verbindung (CONNAUTH) eine Zeichenfolge, die mit der AUTHINFO-Definition übereinstimmt.

CHCKLOCL akzeptiert die Werte NONE und OPTIONALund **CHCKCLNT** ermöglicht die Konfiguration des Werts NONE für die Authentifizierungsanforderungen:

Ohne
: Schaltet die Überprüfung aus.

OPTIONAL
Stellt bei der Bereitstellung einer Benutzer-ID und eines Kennworts durch eine Anwendung sicher, dass es sich um ein gültiges Paar handelt, diese Bereitstellung jedoch nicht obligatorisch ist. Diese Option kann beispielsweise bei einer Migration hilfreich sein.

Wichtig: OPTIONAL ist der Mindestwert, den Sie festlegen können, um die strengeren CHLAUTH-Regeln zu verwenden.

Wenn Sie NONE auswählen und die Clientverbindung mit einem CHLAUTH-Datensatz mit dem Wert CHCKCLNT REQUIRED (oder REQDADM auf anderen Plattformen als z/OS) übereinstimmt, schlägt die Verbindung fehl. Sie erhalten die Nachricht AMQ9793 auf anderen Plattformen als z/OS und die Nachricht CSQX793E unter z/OS.

erforderlich
Alle Anwendungen müssen eine gültige Benutzer-ID und ein gültiges Kennwort bereitstellen. Siehe auch den folgenden Hinweis.

REQDADM
Privilegierte Benutzer müssen eine gültige Benutzer-ID und ein gültiges Kennwort bereitstellen, aber nicht privilegierte Benutzer werden wie bei der Einstellung OPTIONAL behandelt. Siehe auch den folgenden Hinweis. z/OS (Diese Einstellung ist auf z/OS-Systemen nicht zulässig.)

Anmerkung:

Wenn Sie **CHCKLOCL** auf REQUIRED oder REQDADM setzen, können Sie den Warteschlangenmanager nicht lokal verwalten, indem Sie **runmqsc** (Fehler AMQ8135: Nicht autorisiert) verwenden, es sei denn, der Benutzer gibt den Parameter -u UserId in der **runmqsc** -Befehlszeile an. Mit dieser Gruppe fordert **runmqsc** zur Eingabe des Benutzerkennworts an der Konsole auf.

Entsprechend wird einem Benutzer, der IBM MQ Explorer auf dem lokales ´n System ausführt, der Fehler AMQ4036 angezeigt, wenn er eine Verbindung mit dem Warteschlangenmanager herstellen will. Um einen Benutzernamen und ein Kennwort anzugeben, klicken Sie mit der rechten Maustaste auf das lokale Warteschlangenmanagerobjekt und wählen Sie **Verbindungsdetails** > **Eigenschaften ...** aus. aus dem Menü. Geben Sie im Abschnitt **Benutzer-ID** den Benutzernamen und das Kennwort ein, die verwendet werden sollen, und klicken Sie dann auf **OK** .

Ähnliche Hinweise gelten für ferne Verbindungen mit **CHCKCLNT**.

CONNAUTH ist für migrierte WS-Manager leer, aber für neue Warteschlangenmanager auf *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* gesetzt. In der vorhergehenden **AUTHINFO** -Definition ist **CHCKCLNT** standardmäßig auf *REQDADM* gesetzt.

Aus diesem Grund müssen Sie das richtige Betriebssystemkennwort für alle vorhandenen Clients angeben, die eine privilegierte Benutzer-ID verwenden, um eine Verbindung herzustellen.

Warnung: In einigen Fällen wird das Kennwort in einer MQCSP-Struktur für eine Clientanwendung über ein Netz in Klartext gesendet. Die Informationen im Abschnitt „MQCSP-Kennwortschutz” auf Seite 34 erläutern, wie Sie sicherstellen können, dass Clientanwendungskennwörter angemessen geschützt sind.

Konfigurationsgranularität

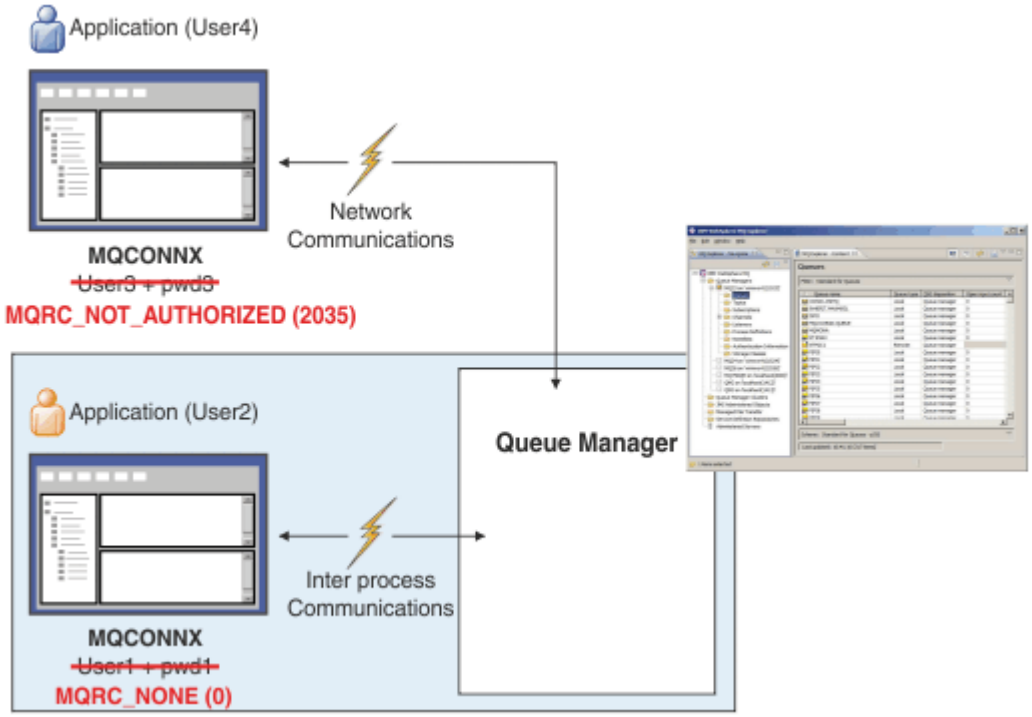
Zusätzlich zu **CHCKLOCL** und **CHCKCLNT** , die zum Aktivieren der Benutzer-ID- und Kennwortprüfung verwendet werden, gibt es Erweiterungen an den CHLAUTH -Regeln, sodass eine spezifischere Konfiguration mit **CHCKCLNT**vorgenommen werden kann.

Sie können den Gesamtwert **CHCKCLNT** beispielsweise auf OPTIONALsetzen und dann ein Upgrade durchführen, damit er für bestimmte Kanäle strenger ist, indem Sie **CHCKCLNT** in der Regel CHLAUTH auf ERFORDERLICH oder REQDADM setzen. Standardmäßig werden CHLAUTH -Regeln mit CHCKCLNT(ASQMGR) ausgeführt, sodass diese Granularität nicht verwendet werden muss. For example:

```
DEFINE AUTHINFO(USE.PW) AUTHTYPE(xxxxxx) +
CHCKCLNT(OPTIONAL)
SET CHLAUTH('*') TYPE(ADDRESSMAP) +
ADDRESS('*') USERSRC(CHANNEL) +
```

```
CHCKCLNT(REQUIRED)
SET CHLAUTH('*') TYPE(SSLPEERMAP) +
SSLPEER('CN=*') USERSRC(CHANNEL)
```

Fehlerbenachrichtigung



Ein Fehler wird aufgezeichnet, wenn eine Anwendung bei Bedarf keine Benutzer-ID und kein Kennwort zur Verfügung stellt oder eine falsche Kombination auch dann bereitstellt, wenn sie optional ist.

Anmerkung: Wenn die Kennwortprüfung inaktiviert ist, werden ungültige Kennwörter nicht erkannt, indem Sie die Option NONE in **CHCKLOCL** oder **CHCKCLNT** verwenden.

Fehlgeschlagene Authentifizierungen werden für die vom Attribut **FAILDLAY** angegebene Anzahl von Sekunden angehalten, bevor der Fehler an die Anwendung zurückgegeben wird. Dies bietet einen gewissen Schutz von einer Anwendung, die wiederholt versucht, eine Verbindung herzustellen.

Der Fehler wird auf eine Reihe von Arten aufgezeichnet:

Anwendung
Die Anwendung gibt den standardmäßigen IBM MQ-Sicherheitsfehler RC2035 - MQRC_NOT_AUTHORIZED zurück.

Administrator
Ein IBM MQ-Administrator kann das im Fehlerprotokoll dokumentierte Ereignis anzeigen und somit feststellen, dass die Anwendung abgelehnt wurde, weil die Benutzer-ID und das Kennwort die Prüfung nicht bestanden haben und nicht, weil beispielsweise keine Verbindungberechtigung vorhanden war.

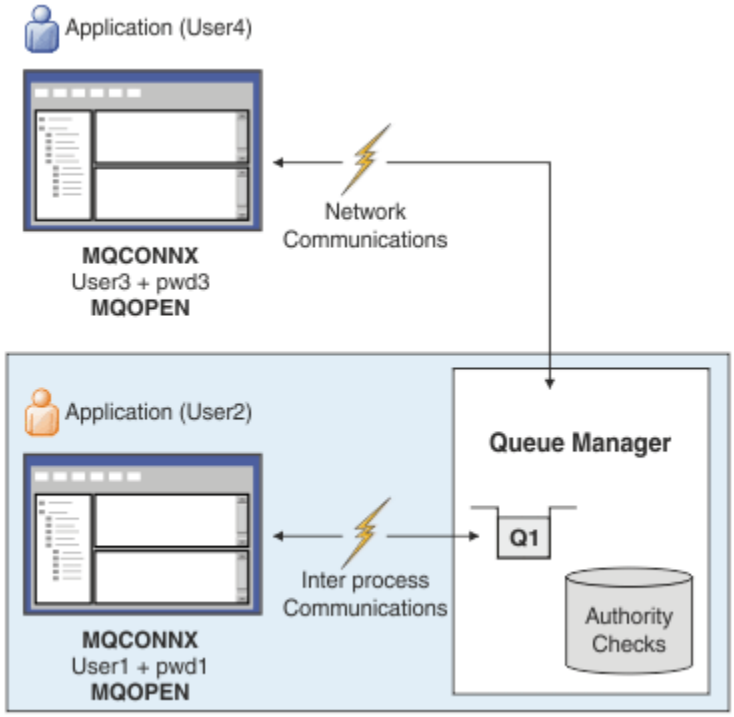
Überwachungstool
Ein Überwachungstool kann auch über den Fehler informiert werden, wenn Sie Berechtigungsereignisse aktivieren, indem Sie eine Ereignisnachricht an die Warteschlange SYSTEM.ADMIN.QMGR.EVENT senden:

```
ALTER QMGR AUTHOREV(ENABLED)
```

Dieses Ereignis "Nicht berechtigt" ist ein Verbindungsereignis vom Typ 1 und bietet dieselben Felder wie andere Ereignisse des Typs 1 mit einem zusätzlichen Feld, der MQCSP-Benutzer-ID, die bereitgestellt wurde. Das Kennwort wird in der Ereignisnachricht nicht angegeben. Dies bedeutet, dass in

der Ereignisnachricht zwei Benutzer-IDs vorhanden sind: die ID, unter der die Anwendung ausgeführt wird, und die ID, die die Anwendung für die Überprüfung der Benutzer-ID und des Kennworts vorlegt.

Beziehung zur Berechtigung



Sie können einen Warteschlangenmanager für das Mandat konfigurieren, dass Benutzer-IDs und Kennwörter von bestimmten Anwendungen bereitgestellt werden, da die Benutzer-ID, unter der die Anwendung ausgeführt wird, nicht dieselbe Benutzer-ID ist, die von der Anwendung zusammen mit einem Kennwort angezeigt wurde, wenn die Anwendung eine Warteschlange für die Ausgabe öffnet, z. B.:

```
ALTER QMGR CONNAUTH(USE.PWD)
DEFINE AUTHINFO(USE.PWD) +
AUTHTYPE(xxxxxx) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED) +
ADOPTCTX(YES)
```

Wie Benutzer-IDs und Kennwörter gehandhabt werden, wird durch das Attribut **ADOPTCTX** im Authentifizierungsinformationsobjekt gesteuert.

ADOPTCTX (YES)
Alle Berechtigungsprüfungen für eine Anwendung werden mit derselben Benutzer-ID durchgeführt, die Sie durch das Kennwort authentifiziert haben, indem Sie auswählen, dass der Kontext als Anwendungskontext für den Rest der Lebensdauer der Verbindung übernommen werden soll.

Achtung: Bei Verwendung von ADOPTCTX(YES) und OS-Benutzer-IDs müssen Sie sicherstellen, dass die zu übernehmende Benutzer-ID die maximale Länge der Benutzer-IDs nicht überschreitet. Weitere Informationen finden Sie unter „Benutzer-IDs” auf Seite 93.

ADOPTCTX (NO)
Eine Anwendung stellt eine Benutzer-ID und ein Kennwort zur Verfügung, um sie bei der Verbindungszeit zu authentifizieren, aber dann unter Verwendung der Benutzer-ID, unter der die Anwendung für zukünftige Berechtigungsprüfungen ausgeführt wird, fortgesetzt wird. Sie finden diese Option möglicherweise bei der Migration oder wenn Sie planen, andere Mechanismen, wie z. B. Kanalauthentifizierungs-Aufzeichnungen, zu verwenden, um die Benutzer-ID des Nachrichtenkanalagenten (MCAUSER) zuzuordnen.

Achtung:

Wenn Sie den Parameter **ADOPTCTX(YES)** in einem Authentifizierungsinformationsobjekt verwenden, kann ein anderer Sicherheitskontext nicht übernommen werden, es sei denn, Sie setzen den Parameter **ChlauthEarlyAdopt** in der Zeilengruppe 'channels' der Datei `qm.ini` ein.

Beispiel: Das Standardauthentifizierungsinformationsobjekt wird auf **ADOPTCTX(YES)** gesetzt, und der Benutzer `fred` ist angemeldet. Die folgenden beiden CHLAUTH-Regeln sind konfiguriert:

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

Der folgende Befehl wird ausgegeben, mit dem Zweck, den Befehl als den angenommenen Sicherheitskontext des Benutzers bob zu authentifizieren:

```
runmqsc -c -u bob QMGR
```

In der Tat verwendet der Warteschlangenmanager den Sicherheitskontext von `fred`, nicht bob, und die Verbindung schlägt fehl.

Weitere Informationen zu **ChlauthEarlyAdopt**finden Sie unter Attribute der Zeilengruppe 'channels'.

Zugehörige Konzepte

„Verbindungsauthentifizierung” auf Seite 75

„Verbindungsauthentifizierung: Anwendungsänderungen” auf Seite 80

„Verbindungsauthentifizierung: Benutzerrepositorys” auf Seite 81

Für jeden Ihrer Warteschlangenmanager können Sie verschiedene Typen von Authentifizierungsinformationsobjekten für die Authentifizierung von Benutzer-IDs und Kennwörtern auswählen.

Verbindungsauthentifizierung: Anwendungsänderungen

Eine Anwendung kann eine Benutzer-ID und ein Kennwort in der MQCSP-Struktur (MQCSP = Verbindungssicherheitsparameter) bereitstellen, wenn MQCONNX aufgerufen wird. Die Benutzer-ID und das Kennwort werden zur Prüfung an den Objektberechtigungsmanager (OAM), der mit dem Warteschlangenmanager geliefert wird, oder an die Berechtigungsservicekomponente übergeben, die mit dem Warteschlangenmanager auf z/OS-Systemen bereitgestellt wird. Sie müssen Ihre eigene angepasste Schnittstelle nicht schreiben.

Wenn die Anwendung als Client ausgeführt wird, werden die Benutzer-ID und das Kennwort auch zur Verarbeitung an die clientseitigen und serverseitigen Sicherheitsexits übergeben. Sie können auch verwendet werden, um das Attribut user identifier (MCAUSER) des Nachrichtenkanalagenten einer Kanalinstanz festzulegen. Der Sicherheitsexit wird mit der Exitursache MQXR_SEC_PARMS für diese Verarbeitung aufgerufen. Die clientseitigen Sicherheitsexits und der Exit für die Vorabverbindung können Änderungen an MQCONN vornehmen, bevor sie an den Queue Manager gesendet werden.

Warnung: In einigen Fällen wird das Kennwort in einer MQCSP-Struktur für eine Clientanwendung über ein Netz in Klartext gesendet. Um sicherzustellen, dass die Kennwörter der Clientanwendung ordnungsgemäß geschützt sind, finden Sie weitere Informationen in „MQCSP-Kennwortschutz” auf Seite 34.

Wenn Sie die Zeichenfolge "XAOPEN" verwenden, um eine Benutzer-ID und ein Kennwort bereitzustellen, können Sie vermeiden, dass Änderungen am Anwendungscode vorgenommen werden müssen.

Anmerkung:

Ab IBM WebSphere MQ 6.0 ermöglicht der Sicherheitsexit die Festlegung von MQCSP. Daher müssen Clients auf dieser Ebene oder höher nicht aktualisiert werden.

In Versionen von IBM MQ vor IBM MQ 8.0 gab es von MQCSP keine Einschränkungen für die Benutzer-ID und das Kennwort, die mit der Anwendung bereitgestellt wurden. Wenn Sie diese Werte mit den von IBM MQ bereitgestellten Features verwenden, gibt es Grenzwerte, die für die Verwendung dieser Features gelten. Wenn Sie sie jedoch nur an Ihre eigenen Exits übergeben, gelten diese Einschränkungen nicht.

Zugehörige Konzepte

„Verbindungsauthentifizierung” auf Seite 75

„Verbindungsauthentifizierung: Konfiguration” auf Seite 76
Ein Warteschlangenmanager kann so konfiguriert werden, dass er eine angegebene Benutzer-ID und ein Kennwort verwendet, um zu überprüfen, ob ein Benutzer über die Berechtigung für den Zugriff auf Ressourcen verfügt.

„Verbindungsauthentifizierung: Benutzerrepositorys” auf Seite 81
Für jeden Ihrer Warteschlangenmanager können Sie verschiedene Typen von Authentifizierungsinformationsobjekten für die Authentifizierung von Benutzer-IDs und Kennwörtern auswählen.

Verbindungsauthentifizierung: Benutzerrepositorys

Für jeden Ihrer Warteschlangenmanager können Sie verschiedene Typen von Authentifizierungsinformationsobjekten für die Authentifizierung von Benutzer-IDs und Kennwörtern auswählen.

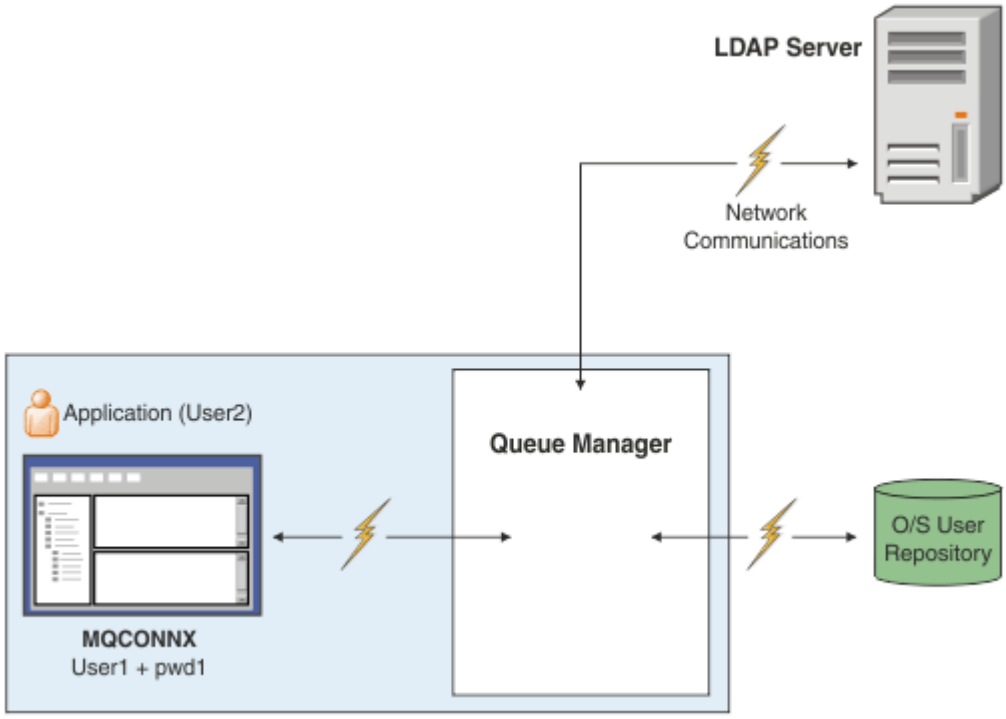


Abbildung 7. Typen von Authentifizierungsinformationsobjekten

```
DEFINE AUTHINFO(USE.OS) AUTHTYPE(IDPWOS)
DEFINE AUTHINFO(USE.LDAP) +
AUTHTYPE(IDPWLDAP) +
CONNAME('ldap1(389),ldap2(389)') +
LDAPUSER('CN=QMGR1') +
LDAPPWD('passw0rd') SECCOMM(YES)
```

Es gibt zwei Typen von Authentifizierungsinformationsobjekten, die im Diagramm dargestellt werden:

- Mit `IDPWOS` wird angegeben, dass der Warteschlangenmanager das lokale Betriebssystem verwendet, um die Benutzer-ID und das Kennwort zu authentifizieren. Wenn Sie sich für die Verwendung des lokalen Betriebssystems entscheiden, müssen Sie die allgemeinen Attribute wie in den vorherigen Abschnitten beschrieben definieren.
- `IDPWLDAP` wird verwendet, um anzugeben, dass der Warteschlangenmanager einen LDAP-Server verwendet, um die Benutzer-ID und das Kennwort zu authentifizieren. Wenn Sie einen LDAP-Server verwenden möchten, finden Sie weitere Informationen in diesem Thema.

Für jeden zu verwendenden Warteschlangenmanager kann nur ein Typ von Authentifizierungsinformationsobjekt ausgewählt werden, indem das entsprechende Objekt im Attribut **CONNAUTH** des WS-Managers angegeben wird.

Verwendung eines LDAP-Servers für die Authentifizierung.

Setzen Sie das Feld **CONNAME** auf die Adresse des LDAP-Servers für den Warteschlangenmanager. Sie können mehr Adressen für den LDAP-Server in einer durch Kommas getrennten Liste angeben, die bei der Redundanz hilfreich sein kann, wenn der LDAP-Server diese Funktion nicht selbst bereitstellt.

Legen Sie die erforderliche LDAP-Server-ID und das erforderliche Kennwort in den Feldern **LDAPUSER** und **LDAPPWD** fest, damit der WS-Manager auf den LDAP-Server zugreifen und Informationen zu Benutzerdatensätzen suchen kann.

Sichere Verbindung zu einem LDAP-Server

Im Gegensatz zu Kanälen gibt es keinen **SSLCIPH** -Parameter, um die Verwendung von TLS für die Kommunikation mit dem LDAP-Server zu aktivieren. In diesem Fall dient IBM MQ als Client für den LDAP-Server, so dass ein Großteil der Konfiguration auf dem Konfiguration wird. Einige vorhandene Parameter in IBM MQ werden dazu verwendet, die Funktionsweise dieser Verbindung zu konfigurieren.

Legen Sie das Feld **SECCOMM** fest, um zu steuern, ob die Verbindung zum LDAP-Server TLS verwendet.

In addition to this attribute, the queue manager attributes **SSLFIPS** and **SUITEB** restrict the set of cipher specs that are chosen. Das Zertifikat, das zum Identifizieren des Warteschlangenmanagers für den LDAP-Server verwendet wird, ist das WS-Manager-Zertifikat, entweder `ibmwebspheremq` *`qmgr-name`* oder der Wert des Attributs **CERTLABL** . Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

LDAP-Benutzerrepository

Bei Verwendung eines LDAP-Benutzerrepositorys gibt es eine weitere Konfiguration, die auf dem WS-Manager ausgeführt werden muss, als nur dem Warteschlangenmanager mitzuteilen, wo der LDAP-Server zu finden ist.

Die in einem LDAP-Server definierten Benutzer-IDs verfügen über eine hierarchische Struktur, die sie eindeutig identifiziert. Daher kann eine Anwendung eine Verbindung zum WS-Manager herstellen und ihre Benutzer-ID als vollständig qualifizierte hierarchische Benutzer-ID darstellen.

Um jedoch die Informationen zu vereinfachen, die eine Anwendung bereitstellen muss, ist es möglich, den Warteschlangenmanager so zu konfigurieren, dass der erste Teil der Hierarchie allen IDs gemeinsam ist, und diese vor der gekürzten ID, die von der Anwendung bereitgestellt wird, automatisch hinzufügen. Der WS-Manager kann dann eine vollständige ID für den LDAP-Server darstellen.

Setzen Sie BASEDNU auf den Anfangspunkt, den die LDAP-Suche nach der ID in der LDAP-Hierarchie sucht. Wenn Sie BASEDNU festlegen, müssen Sie sicherstellen, dass bei der Suche nach der ID in der LDAP-Hierarchie nur ein einziges Ergebnis zurückgegeben wird.

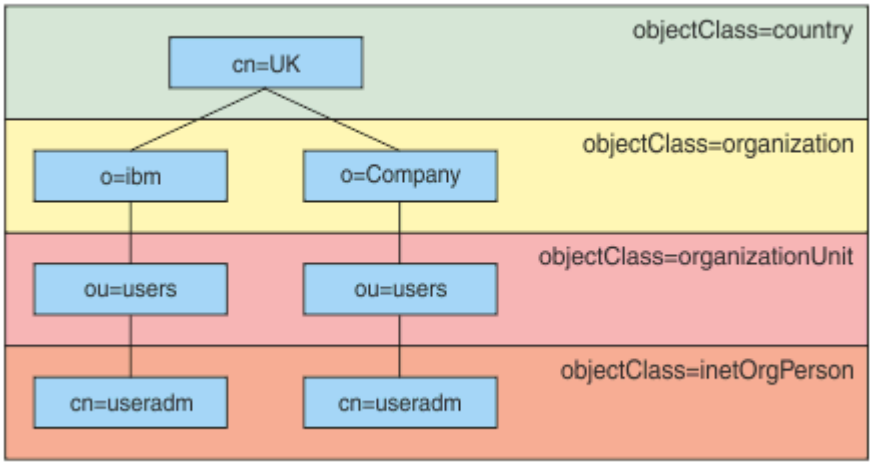


Abbildung 8. Beispiel einer LDAP-Hierarchie

Beispiel: In Abbildung 8 auf Seite 83 kann BASEDNU auf "ou=users,o=ibm,c=UK" oder ",o=ibm,c=UK" gesetzt werden. Da jedoch ein definierter Name, der "cn = useradm" enthält, sowohl in der Verzweigung "o = ibm" als auch in der Verzweigung "o=Unternehmen" vorhanden ist, kann BASEDNU nicht auf "c = UK" gesetzt werden. Verwenden Sie für Leistungs-und Sicherheitsgründe den höchsten Punkt in Ihrer LDAP-Hierarchie, von dem aus Sie alle benötigten Benutzer-IDs referenzen können. In diesem Beispiel ist dies "ou=users, o=ibm, c = UK".

Ihre Anwendung kann die Benutzer-ID unter Umständen an den Warteschlangenmanager übergeben, ohne den LDAP-Attributnamen, z. B. CN= , bereitzustellen. Wenn Sie USRFIELD auf den LDAP-Attributnamen setzen, wird dieser Wert als Präfix zu der Benutzer-ID hinzugefügt, die aus der Anwendung stammt. Dies kann eine nützliche Migrationshilfe sein, wenn Sie von Betriebssystembenutzer-IDs zu LDAP-Benutzer-IDs wechseln, da die Anwendung dann in beiden Fällen dieselbe Zeichenfolge darstellen kann und Sie die Änderung der Anwendung vermeiden können.

Daher sieht die vollständige Benutzer-ID, die dem LDAP-Server angezeigt wird, wie folgt aus:

```
USRFIELD = ID_from_application BASEDNU
```

Zugehörige Konzepte

„Verbindungsauthentifizierung” auf Seite 75

„Verbindungsauthentifizierung: Konfiguration” auf Seite 76
Ein Warteschlangenmanager kann so konfiguriert werden, dass er eine angegebene Benutzer-ID und ein Kennwort verwendet, um zu überprüfen, ob ein Benutzer über die Berechtigung für den Zugriff auf Ressourcen verfügt.

„Verbindungsauthentifizierung: Anwendungsänderungen” auf Seite 80

Clientseitiger Sicherheitsexit zum Einfügen von Benutzer-ID und Kennwort (`mqccred`)

Wenn Sie über Clientanwendungen verfügen, die zum Senden einer Benutzer-ID oder eines Kennworts erforderlich sind, aber Sie die Quelle noch nicht ändern können, wird ein Sicherheitsexit mit dem IBM MQ 8.0 , mit dem Namen **`mqccred`** geliefert, den Sie verwenden können. **`mqccred`** stellt eine Benutzer-ID und ein Kennwort für die Clientanwendung aus einer `.ini` -Datei bereit. Diese Benutzer-ID und dieses Kennwort werden an den Warteschlangenmanager gesendet, der sie authentifizieren wird, wenn dies entsprechend konfiguriert ist.

Übersicht

`mqccred` ist ein Sicherheitsexit, der auf derselben Maschine wie Ihre Clientanwendung ausgeführt wird. Sie ermöglicht die Angabe von Benutzerkennungs-und Kennwortinformationen im Namen der Clientanwendung, wenn diese Informationen nicht von der Anwendung selbst bereitgestellt werden. Die Informa-

tionen zur Benutzer-ID und zum Kennwort werden in einer Struktur bereitgestellt, die als Parameter für Verbindungssicherheitsparameter (MQCSP) bezeichnet wird, und wird vom Warteschlangenmanager authentifiziert, wenn die Verbindungsauthentifizierung konfiguriert ist.

Benutzer-ID-und Kennwortinformationen werden aus einer `.ini` -Datei auf der Clientmaschine abgerufen. Die Kennwörter in der Datei werden durch Verschlüsselung mit dem Befehl **runmqccred** geschützt. Außerdem wird sichergestellt, dass die Dateiberechtigungen für die Datei `.ini` so festgelegt werden, dass nur die Benutzer-ID, die die Clientanwendung (und damit den Exit) ausführt, sie lesen kann.

Position

mqccred ist installiert:

Windows-Plattformen
Im *installation_directory*\Tools\c\Samples\mqccred\ Verzeichnis

AIX and Linux-Plattformen
Im *installation_directory*/samp/mqccred Verzeichnis

Anmerkungen: Der Exit:

1. Ist nur als Sicherheitskanalexit aktiv und muss der einzige in einem Kanal definierte Exit sein.
2. Wird normalerweise über die Definitionstabelle für den Clientkanal (CCDT) benannt, aber ein Java-Client kann den in den JNDI-Objekten direkt angegebenen Exit haben oder der Exit kann für Anwendungen konfiguriert werden, die die MQCD-Struktur manuell erstellen.
3. Sie müssen die Programme **mqccred** und **mqccred_r** in das Verzeichnis `var/mqm/exits` kopieren.

   Geben Sie zum Beispiel auf einem AIX- oder Linux-System mit 64 Bit den folgenden Befehl aus:

   ```
   cp installation_directory/samp/mqccred/lib64/* /var/mqm/exits
   ```

   Weitere Informationen finden Sie im Abschnitt Schritt für Schritt zum Testen von 'mqccred' .
4. Kann in früheren Versionen von IBM MQ ausgeführt werden (bis IBM WebSphere MQ 7.0.1).

Benutzer-IDs und Kennwörter konfigurieren

Die Datei `.ini` enthält Zeilengruppen für jeden Warteschlangenmanager und enthält eine globale Einstellung für nicht angegebene Warteschlangenmanager. Jede Zeilengruppe enthält den Namen des Warteschlangenmanagers, eine Benutzer-ID und entweder ein Klartext oder ein Kennwort mit einer Kennung.

Sie müssen die `.ini` -Datei manuell bearbeiten, indem Sie den gewünschten Editor verwenden und den Zeilengruppen das Attribut "Klartextkennwort" hinzufügen. Führen Sie das bereitgestellte Programm **runmqccred** aus, das die Datei `.ini` verwendet und das Attribut **Password** durch das Attribut **OPW** in verschlüsselter Form des Kennworts ersetzt.

Eine Beschreibung des Befehls und seiner Parameter finden Sie in runmqccred .

Die Datei `mqccred.ini` enthält Ihre Benutzer-ID-und Kennwortinformationen.

Eine Schablonendatei `.ini` wird im selben Verzeichnis wie der Exit bereitgestellt, um einen Ausgangspunkt für Ihr Unternehmen bereitzustellen.

Standardmäßig wird diese Datei in $HOME/`.mqs/mqccred.ini` gesucht. Wenn Sie sie an anderer Stelle suchen möchten, können Sie die Umgebungsvariable *MQCCRED* verwenden, um auf sie zu verweisen:

```
MQCCRED=C:\mydir\mqccred.ini
```

Wenn Sie *MQCCRED* verwenden, muss die Variable den vollständigen Namen der Konfigurationsdatei enthalten, einschließlich aller `.ini` -Dateitypen. Da diese Datei Kennwörter enthält (auch wenn sie von der Verschlüsselung entfernt wird), müssen Sie die Datei mit Hilfe von Betriebssystemberechtigungen schützen, um sicherzustellen, dass nicht autorisierte Personen diese Datei nicht lesen können. Wenn Sie nicht über die korrekte Dateiberechtigung verfügen, wird der Exit nicht erfolgreich ausgeführt.

Wenn die Anwendung bereits eine MQCSP -Struktur angegeben hat, respektiert der Exit normalerweise diese und fügt keine Informationen aus der `.ini` -Datei ein. Sie können diese Eigenschaft jedoch mit dem Attribut **Force** in der Zeilengruppe überschreiben.

Wenn Sie **Force** auf den Wert *TRUE* setzen, wird die von der Anwendung bereitgestellte Benutzer-ID und das Kennwort entfernt, und diese werden durch die Version der INI-Datei ersetzt.

Sie können auch das Attribut **Force** im globalen Abschnitt der Datei festlegen, um den Standardwert für diese Datei festzulegen.

Der Standardwert für **Force** ist *FALSE*.

Sie können eine Benutzer-ID und ein Kennwort für alle Warteschlangenmanager oder für jeden einzelnen WS-Manager angeben. Dies ist ein Beispiel für eine `mqccred.ini` -Datei:

```
# comments are permitted
AllQueueManagers:
User=abc
OPW=%^&aervrgtsr

QueueManager:
Name=QMA
User=user1
OPW=H&^dbgfh

Force=TRUE

QueueManager:
Name=QMB
User=user2
password=passw0rd
```

Anmerkungen:

1. Die einzelnen WS-Manager-Definitionen haben Vorrang vor der globalen Einstellung.
2. Bei Attributen wird die Groß-/Kleinschreibung nicht beachtet.

Integritätsbedingungen

Wenn dieser Exit im Gebrauch ist, wird die lokale Benutzer-ID der Person, die die Anwendung ausführt, nicht vom Client zum Server fließen. Die einzigen verfügbaren Identitätsinformationen sind aus dem Inhalt der INI-Datei.

Aus diesem Grund müssen Sie den Warteschlangenmanager so konfigurieren, dass er entweder **ADOPTCTX(YES)** verwendet, oder die eingehende Verbindungsanforderung über einen der verfügbaren Mechanismen (z. B. „Kanalauthentifizierungssätze” auf Seite 54) einer entsprechenden Benutzer-ID zuordnen.

Wichtig: Wenn Sie neue Kennwörter hinzufügen oder alte Kennwörter aktualisieren, verarbeitet der Befehl **runmqccred** nur Klartextkennwörter und lässt Ihre verschlüsselten Kennwörter unberührt.

Fehlerbehebung

Der Exit schreibt in den IBM MQ-Standardtrace, wenn dieser aktiviert ist.

Zur Unterstützung beim Debugging von Konfigurationsproblemen kann der Exit auch direkt in stdout schreiben.

Für den Kanal ist normalerweise keine Konfiguration der Kanalsicherheitsexit-Daten (**SCYDATA**) erforderlich. Sie können jedoch Folgendes angeben:

FEHLER
Es werden nur Fehlerbedingungen für die Druckinformationen ausgegeben, z. B. wenn die Konfigurationsdatei nicht gefunden werden kann.

DEBUG
Zeigt diese Fehlerbedingungen und einige zusätzliche Traceanweisungen an.

NOCHECKS
Umgeht die Einschränkungen für Dateiberechtigungen und die weitere Einschränkung, dass die Datei `.ini` keine ungeschützten Kennwörter enthalten sollte.

Sie können eines oder mehrere dieser Elemente in das Feld **SCYDATA** (durch Kommas getrennt) in beliebiger Reihenfolge einlegen. Beispiel: `SCYDATA=(NOCHECKS,DEBUG)`.

Beachten Sie, dass bei den Elementen die Groß-/Kleinschreibung beachtet werden muss und dass sie in Großbuchstaben eingegeben werden müssen.

`mqccred` verwenden

Sobald die Datei eingerichtet ist, können Sie den Kanalexit aufrufen, indem Sie Ihre Clientverbindungskanaldefinition so aktualisieren, dass sie das Attribut `SCYEXIT('mqccred(ChlExit)')` enthält:

```
DEFINE CHANNEL(channelname) CHLTYPE(clntconn) +
CONNAME(remote machine) +
QMNAME(remote qmgr) +
SCYEXIT('mqccred(ChlExit)') +
REPLACE
```

Zugehörige Verweise
SCYDATA
SCYEXIT
runmqccred

Verbindungsauthentifizierung mit dem Java-Client

Bei der Verbindungsauthentifizierung handelt es sich um eine Funktion in IBM MQ, die es Ihnen ermöglicht, Warteschlangenmanager so zu konfigurieren, dass der Warteschlangenmanager Anwendungen mit einer bereitgestellten Benutzer-ID und einem angegebenen Kennwort authentifizieren kann. Wenn es sich bei der Anwendung um eine Java-Anwendung handelt, die den Clienttransport verwendet, kann die Verbindungsauthentifizierung im Kompatibilitätsmodus oder im MQCSP-Authentifizierungsmodus ausgeführt werden.

Die Benutzer-ID und das Kennwort, die authentifiziert werden sollen, werden von der Anwendung mit einer der folgenden Methoden angegeben:

- In einer IBM MQ classes for Java-Anwendung, in der Klasse `MQEnvironment` oder in der Eigenschaft Hashtable, die an den Konstruktor `com.ibm.mq.MQQueueManager` übergeben wird.
- In einer IBM MQ classes for JMS -Anwendung als Argumente für die Methode `createConnection(String username, String Password)` oder `createContext(String username, String password)` .

MQCSP-Authentifizierungsmodus

In diesem Modus werden die clientseitige Benutzer-ID, unter der die Anwendung ausgeführt wird, an den Warteschlangenmanager gesendet, sowie die Benutzer-ID und das Kennwort, die authentifiziert werden sollen. IBM MQ classes for Java und IBM MQ classes for JMS senden die Benutzer-ID und das Kennwort, die authentifiziert werden sollen, an den Warteschlangenmanager in einer MQCSP-Struktur.

Die Benutzer-ID und das Kennwort sind für einen Serververbindungssicherheitsexit in der MQCSP-Struktur verfügbar. Die Adresse der MQCSP-Struktur ist im Feld **SecurityParms**der MQCXP-Struktur für den Kanal zu finden.

Der MQCSP-Authentifizierungsmodus hat die folgenden Vorteile:

- Die maximale Länge der zu authentifizierenden Benutzer-ID beträgt 1024 Zeichen.
- Die maximale Länge des Kennworts für die Authentifizierung beträgt 256 Zeichen.
- Berechtigungsprüfungen für den Zugriff auf die Verwendung von IBM MQ-Ressourcen können mit der clientseitigen Benutzer-ID ausgeführt werden, unter der die Anwendung ausgeführt wird, wenn

das Authentifizierungsinformationsobjekt, das zur Steuerung der Verbindungsauthentifizierung auf dem Warteschlangenmanager verwendet wird, mit ADOPTCTX (NO) konfiguriert ist.

Kompatibilitätsmodus

Vor IBM MQ 8.0 konnte der Java-Client eine Benutzer-ID und ein Kennwort über den Clientverbindungskanal an den Serververbindungskanal senden und die Informationen einem Sicherheitsexit in den Feldern **RemoteUserIdentifier** und **RemotePassword** der MQCD-Struktur bereitstellen. Im Kompatibilitätsmodus wird dieses Verhalten beibehalten.

Sie können diesen Modus in Kombination mit der Verbindungsauthentifizierung verwenden und von allen Sicherheitsexits migrieren, die zuvor für denselben Job verwendet wurden.

Dieser Modus hat die folgenden Einschränkungen:

- Die Länge der Benutzer-ID und des Kennworts muss 12 Zeichen oder weniger sein. Benutzer-IDs, die länger als 12 Zeichen sind, werden auf 12 Zeichen abgeschnitten. Dies kann dazu führen, dass die Verbindung mit dem Ursachencode MQRC_NOT_AUTHORIZED fehlschlägt.
- Die clientseitige Benutzer-ID, unter der die Anwendung ausgeführt wird, wird nicht an den Warteschlangenmanager gesendet. Sie müssen entweder ADOPTCTX (YES) für das Authentifizierungsinformationsobjekt festlegen, mit dem die Verbindungsauthentifizierung auf dem Warteschlangenmanager gesteuert wird, oder eine andere Methode verwenden, wie z. B. eine Kanalauthentifizierungsregel auf der Basis eines TLS-Zertifikats, um die Kanal-MCA-Benutzer-ID festzulegen, die auf die Berechtigung zur Verwendung von IBM MQ-Ressourcen überprüft wird.

Standardauthentifizierungsmodus

Der Standardauthentifizierungsmodus, der von einer IBM MQ classes for Java- oder IBM MQ classes for JMS-Clientanwendung verwendet wird, hängt davon ab, ob die Anwendung eine Benutzer-ID und ein Kennwort angibt.

- V 9.2.1 Wenn eine Benutzer-ID und ein Kennwort angegeben sind, wird ab IBM MQ 9.2.1 die MQCSP-Authentifizierung standardmäßig verwendet.
- Wenn eine Benutzer-ID und ein Kennwort in früheren Versionen als IBM MQ 9.2.1 angegeben sind, lautet der Standardmodus wie folgt:
  - Die MQCSP-Authentifizierung wird standardmäßig von Anwendungen verwendet, die IBM MQ classes for Java verwenden.
  - Der Kompatibilitätsmodus wird standardmäßig von Anwendungen verwendet, die IBM MQ classes for JMS verwenden.
- Wenn eine Benutzer-ID, aber kein Kennwort angegeben ist, wird standardmäßig der Kompatibilitätsmodus verwendet.
- Wenn keine Benutzer-ID angegeben ist, wird der Kompatibilitätsmodus immer verwendet.

In Fällen, in denen eine Benutzer-ID angegeben ist, kann ein bestimmter Authentifizierungsmodus von der Anwendung für jede einzelne Verbindung ausgewählt oder global festgelegt werden, bevor die Anwendung gestartet wird, wie im Abschnitt „Authentifizierungsmodus auswählen” auf Seite 88 beschrieben.

Anmerkung: V 9.2.1 Anwendungen, die IBM MQ classes for JMS verwenden, können von der Änderung des Standardauthentifizierungsmodus in IBM MQ 9.2.1 betroffen sein. Nach dem Upgrade von IBM MQ classes for JMS auf IBM MQ 9.2.1 werden Anwendungen, die zuvor standardmäßig den Kompatibilitätsmodus verwendet haben, stattdessen die MQCSP-Authentifizierung verwenden. Dies kann dazu führen, dass Anwendungen, die sich zuvor erfolgreich mit einem Warteschlangenmanager verbunden haben, keine Verbindung mehr mit einer `JMSException` herstellen können und den Ursachencode 2035 (MQRC_NOT_AUTHORIZED) ausgeben. Wenn dies der Fall ist, verwenden Sie eine der in „Authentifizierungsmodus auswählen” auf Seite 88 beschriebenen Methoden, um anzugeben, dass die Anwendung den Kompatibilitätsmodus verwendet.

Java-Anwendungen, die mit lokalen Bindungen eine Verbindung zum Warteschlangenmanager herstellen, verwenden immer den MQCSP-Authentifizierungsmodus.

Authentifizierungsmodus auswählen

Der Authentifizierungsmodus, der von Java-Clientanwendungen verwendet wird, die eine Benutzer-ID angeben, wenn eine Verbindung zum Warteschlangenmanager hergestellt werden kann, kann mit einer der folgenden Methoden angegeben werden. Diese Methoden werden in absteigender Reihenfolge aufgelistet. Wenn der Authentifizierungsmodus unter Verwendung einer dieser Methoden nicht angegeben wird, wird der Standardauthentifizierungsmodus verwendet.

Anmerkung: V 9.2.1 Die Verwendung dieser Methoden zur Auswahl des Authentifizierungsmodus wurde in IBM MQ 9.2.1 erläutert. In einigen Fällen kann sich der Authentifizierungsmodus, der von einer Java-Clientanwendung verwendet wird, ändern, wenn für IBM MQ classes for Java oder IBM MQ classes for JMS ein Upgrade auf IBM MQ 9.2.1 durchgeführt wird. Dies kann dazu führen, dass Anwendungen, die sich zuvor erfolgreich mit einem Warteschlangenmanager verbunden haben, keine Verbindung mehr mit einer `JMSException` herstellen können und den Ursachencode 2035 (MQRC_NOT_AUTHORIZED) ausgeben. Wenn dies der Fall ist, verwenden Sie eine der folgenden Methoden, um den Authentifizierungsmodus auszuwählen, der erforderlich ist.

- Geben Sie den Authentifizierungsmodus für jede einzelne Verbindung an, indem Sie die entsprechende Eigenschaft in der Anwendung festlegen, bevor Sie eine Verbindung zum Warteschlangenmanager herstellen.
  - Wenn Sie IBM MQ classes for Java verwenden, legen Sie die Eigenschaft *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* in der Eigenschaft Hashtable fest, die an den Konstruktor `com.ibm.mq.MQQueueManager` übergeben wird.
  - Wenn Sie IBM MQ classes for JMS verwenden, legen Sie die Eigenschaft *JmsConstants.USER_AUTHENTICATION_MQCSP* in der entsprechenden Verbindungsfactory fest, bevor Sie die Verbindung erstellen.

  Setzen Sie den Wert dieser Eigenschaften auf einen der folgenden Werte:

  true
  Verwenden Sie den MQCSP-Authentifizierungsmodus beim Authentifizieren mit einem Warteschlangenmanager.

  false
  Verwenden Sie den Kompatibilitätsmodus, wenn Sie sich mit einem Warteschlangenmanager authentifizieren.
- Geben Sie den Authentifizierungsmodus für alle Clientverbindungen an, die von einer Anwendung hergestellt werden, indem Sie die Systemeigenschaft *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java beim Starten der Anwendung festlegen. Setzen Sie den Wert der Eigenschaft auf einen der folgenden Werte:

  Y
  Verwenden Sie den MQCSP-Authentifizierungsmodus beim Authentifizieren mit einem Warteschlangenmanager.

  N
  Verwenden Sie den Kompatibilitätsmodus, wenn Sie sich mit einem Warteschlangenmanager authentifizieren.

  Mit dem folgenden Befehl wird beispielsweise die Eigenschaft festgelegt, um den Kompatibilitätsmodus auszuwählen; er startet eine Java-Anwendung:

  ```
  java -Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=N application_name
  ```
- Geben Sie den Authentifizierungsmodus für alle Clientverbindungen an, die von Anwendungen hergestellt werden, die in derselben Umgebung gestartet wurden, indem Sie die Umgebungsvariable *com.ibm.mq.jmqi.useMQCSPauthentication* in der Umgebung festlegen, in der die Anwendung gestartet wird. Setzen Sie den Wert der Umgebungsvariablen auf einen der folgenden Werte:

Y
Verwenden Sie den MQCSP-Authentifizierungsmodus beim Authentifizieren mit einem Warteschlangenmanager.

N
Verwenden Sie den Kompatibilitätsmodus, wenn Sie sich mit einem Warteschlangenmanager authentifizieren.

- Geben Sie den Authentifizierungsmodus für alle Anwendungen an, die eine bestimmte IBM MQ MQI clientClient-Konfigurationsdatei verwenden, indem Sie das Attribut**useMQCSPauthentication**in der JMQI-Stanza der Client-Konfigurationsdatei angeben. Setzen Sie den Wert des Attributs auf einen der folgenden Werte:

  YES
  Verwenden Sie den MQCSP-Authentifizierungsmodus beim Authentifizieren mit einem Warteschlangenmanager.

  Nein
  Verwenden Sie den Kompatibilitätsmodus, wenn Sie sich mit einem Warteschlangenmanager authentifizieren.

  Weitere Informationen zum Attribut **useMQCSPauthentication** finden Sie in der JMQI-Stanza der Client-Konfigurationsdatei.

Authentifizierungsmodus in IBM MQ Explorer auswählen

IBM MQ Explorer ist eine Java-Anwendung, sodass der Kompatibilitätsmodus und der MQCSP-Authentifizierungsmodus ebenfalls für diese Anwendung anwendbar sind.

Ab IBM MQ 9.1.0 ist der MQCSP-Authentifizierungsmodus die Standardeinstellung. Vor IBM MQ 9.1 ist der Kompatibilitätsmodus die Standardeinstellung.

In Anzeigen, in denen die Benutzerkennung angegeben ist, gibt es ein Kontrollkästchen, mit dem der Kompatibilitätsmodus aktiviert oder inaktiviert werden kann:

- Ab IBM MQ 9.1.0 ist dieses Kontrollkästchen standardmäßig nicht ausgewählt. Um den Kompatibilitätsmodus zu verwenden, wählen Sie dieses Kontrollkästchen aus.
- Vor IBM MQ 9.1.0 ist dieses Kontrollkästchen standardmäßig aktiviert. Wenn Sie die MQCSP-Authentifizierung verwenden möchten, wählen Sie das Kontrollkästchen ab.

Zugehörige Konzepte

„Verbindungsauthentifizierung” auf Seite 75

„Verbindungsauthentifizierung: Anwendungsänderungen” auf Seite 80

„Verbindungsauthentifizierung: Benutzerrepositorys” auf Seite 81
Für jeden Ihrer Warteschlangenmanager können Sie verschiedene Typen von Authentifizierungsinformationsobjekten für die Authentifizierung von Benutzer-IDs und Kennwörtern auswählen.

Nachrichtensicherheit in IBM MQ

Die Nachrichtensicherheit in der IBM MQ-Infrastruktur wird von Advanced Message Security bereitgestellt.

Advanced Message Security (AMS) erweitert die IBM MQ-Sicherheitsservices, um das Signieren und Verschlüsseln von Daten auf Nachrichtenebene bereitzustellen. Die erweiterten Services stellen sicher, dass die Nachrichtendaten nicht geändert wurden, wenn sie ursprünglich in eine Warteschlange gestellt wurden und wenn sie abgerufen werden. Außerdem stellt AMS sicher, dass ein Sender von Nachrichtendaten berechtigt ist, signierte Nachrichten in eine Zielwarteschlange zu stellen.

Zugehörige Konzepte

„Advanced Message Security” auf Seite 636
Advanced Message Security (AMS) ist eine Komponente von IBM MQ, die ein hohes Maß an Schutz für sensible Daten bereitstellt, die über das IBM MQ-Netz fließen, während die Endanwendungen nicht beeinflusst werden.

Sicherheitsanforderungen planen

In dieser Themensammlung finden Sie Informationen zu den Aspekten, die Sie bei der Planung der Sicherheit in einer IBM MQ-Umgebung berücksichtigen müssen.

Sie können IBM MQ für eine Vielzahl von Anwendung auf verschiedenen Plattformen verwenden. Die Sicherheitsanforderungen können für jede Anwendung unterschiedlich sein. Für einige wird die Sicherheit ein kritischer Aspekt sein.

IBM MQ stellt eine Reihe von Sicherheitsservices auf Verbindungsebene bereit, einschließlich der Unterstützung für Transport Layer Security (TLS).

Sie müssen bestimmte Aspekte der Sicherheit berücksichtigen, wenn Sie planen, IBM MQ zu installieren:

- Multi Wenn Sie unter Multiplatforms diese Aspekte ignorieren und nichts tun, können Sie IBM MQ nicht verwenden.
- z/OS Unter z/OS wirkt sich die Nichtbeachtung dieser Aspekte so aus, dass Ihre IBM MQ-Ressourcen nicht geschützt sind. Das bedeutet, dass alle Benutzer auf alle IBM MQ-Ressourcen zugreifen und diese ändern können.

Berechtigung zum Verwalten von IBM MQ

IBM MQ-Administrator benötigen die folgenden Berechtigungen:

- Ausgabe von Befehlen für die Verwaltung von IBM MQ
- Verwenden von IBM MQ Explorer
- IBM i Verwenden von IBM i-Verwaltungsanzeigen und -Befehlen.
- z/OS Verwenden der Operationen und Steuerkonsolen unter z/OS
- z/OS Verwenden des IBM MQ-Dienstprogramms CSQUTIL unter z/OS
- z/OS Zugriff auf Warteschlangenmanagerdatasets unter z/OS

Weitere Informationen finden Sie unter:

- ALW „Berechtigung für die Verwaltung von IBM MQ unter AIX, Linux, and Windows” auf Seite 428
- IBM i „Berechtigung für die Verwaltung von IBM MQ unter IBM i” auf Seite 95
- z/OS „Berechtigung für die Verwaltung von IBM MQ unter z/OS” auf Seite 96

Berechtigung zur Arbeit mit IBM MQ-Objekten

Anwendungen können durch die Ausgabe von MQI-Aufrufen auf die folgenden IBM MQ-Objekte zugreifen:

- Warteschlangenmanager
- Warteschlangen
- Prozesse
- Namenslisten
- Themen

Anwendungen können auf diese IBM MQ-Objekte sowie auf Kanäle und Authentifizierungsinformationsobjekte auch mithilfe von PCF-Befehlen (Programmable Command Format) zugreifen. Diese Objekte können von IBM MQ geschützt werden, weshalb die den Anwendungen zugeordneten Benutzer-IDs die Berechtigung für den Zugriff auf diese Objekte benötigen.

Weitere Informationen finden Sie im Abschnitt „Berechtigungen für Anwendung zur Verwendung von IBM MQ” auf Seite 98.

Kanalsicherheit

Die Benutzer-IDs, die den Nachrichtenkanalagenten (MCAs) zugeordnet sind, benötigen eine Zugriffsberechtigung für verschiedene IBM MQ-Ressourcen. Ein MCA muss beispielsweise in der Lage sein, eine Verbindung zu einem Warteschlangenmanager herzustellen. Wenn es sich um ein sendende MCA handelt, muss es in der Lage sein, die Übertragungswarteschlange für den Kanal zu öffnen. Wenn es sich um einen empfangenden MCA handelt, muss er in der Lage sein, Zielwarteschlangen zu öffnen. Die Benutzer-IDs, die Anwendungen zugeordnet sind, die Kanäle, Kanalinitiatoren und Empfangsprogramme verwalten müssen, benötigen die Berechtigung zur Verwendung der entsprechenden PCF-Befehle. Die meisten Anwendungen benötigen diesen Zugriff jedoch nicht.

Weitere Informationen finden Sie im Abschnitt „Kanalberechtigung” auf Seite 122.

Weitere Überlegungen

Sie müssen die folgenden Sicherheitsaspekte nur berücksichtigen, wenn Sie bestimmte Erweiterungen der IBM MQ-Funktionen oder des Basisprodukts verwenden:

- „Sicherheit für WS-Manager-Cluster” auf Seite 135
- „Sicherheit für IBM MQ-Publish/Subscribe” auf Seite 136
- „Sicherheit für IBM MQ Internet Pass-Thru” auf Seite 137

Planung der Identifikation und Authentifizierung

Entscheiden Sie, welche Benutzer-IDs verwendet werden sollen und wie und auf welchen Ebenen die Authentifizierungssteuerelemente angewendet werden sollen.

Sie müssen entscheiden, wie die Benutzer Ihrer IBM MQ-Anwendungen identifiziert werden sollen, wobei zu berücksichtigen ist, dass unterschiedliche Betriebssysteme Benutzer-IDs unterschiedlicher Länge unterstützen. Sie können Kanalauthentifizierungsdatensätze verwenden, um eine Zuordnung von einer Benutzer-ID zu einer anderen zu verwenden, oder eine Benutzer-ID basierend auf einem Attribut der Verbindung anzugeben. IBM MQ-Kanäle, die TLS verwenden, verwenden digitale Zertifikate für die Identifikation und Authentifizierung. Jedes digitale Zertifikat verfügt über einen registrierten Namen, der anhand von Kanalauthentifizierungsdatensätzen auf bestimmte Identitäten abgebildet werden kann. Außerdem geben Zertifikate einer Zertifizierungsstelle im Schlüsselrepository an, welche digitalen Zertifikate für die Authentifizierung von IBM MQ verwendet werden können. Weitere Informationen finden Sie unter:

- „Zuordnung eines fernen Warteschlangenmanagers zu einer MCAUSER-Benutzer-ID” auf Seite 413
- „Zuordnung einer Clientbenutzer-ID zu einer MCAUSER-Benutzer-ID” auf Seite 414
- „Zuordnen eines SSL-oder TLS-definierten Namens zu einer MCAUSER-Benutzer-ID” auf Seite 414
- „Zuordnen einer IP-Adresse zu einer MCAUSER-Benutzer-ID” auf Seite 417

Authentifizierung für eine Clientanwendung planen

Sie können Authentifizierungssteuerelemente auf vier Ebenen anwenden: auf der Kommunikationsebene, in Sicherheitsexits, mit Kanalauthentifizierungsdatensätzen und in Bezug auf die Identifikation, die an einen Sicherheitsexit übergeben wird.

Es gibt vier Sicherheitsstufen, die berücksichtigt werden müssen. Das Diagramm zeigt einen IBM MQ MQI client, der mit einem Server verbunden ist. Die Sicherheit wird auf vier Ebenen angewendet, wie im folgenden Text beschrieben. MCA ist ein Nachrichtenkanalagent.

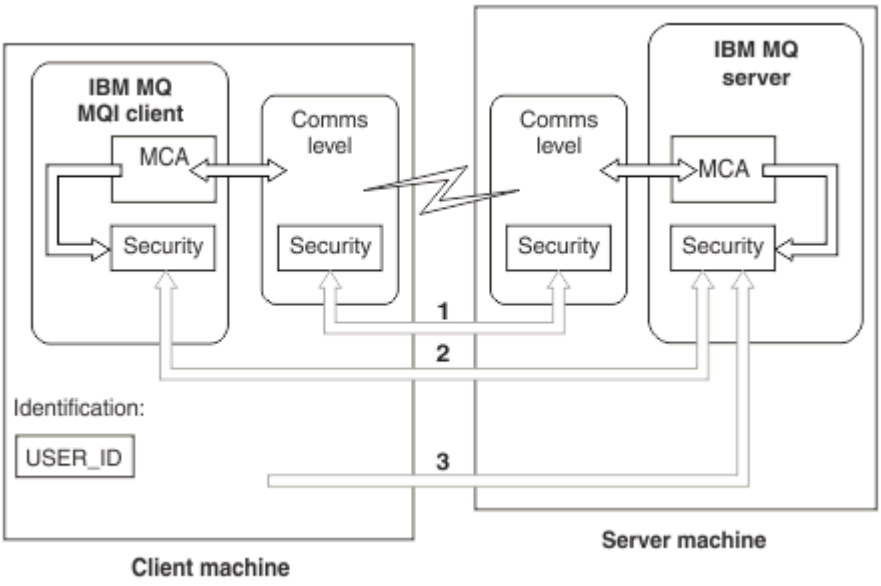


Abbildung 9. Sicherheit in einer Client/Server-Verbindung

1. Übertragungsstufe

   Siehe Pfeil 1. Um die Sicherheit auf Kommunikationsebene zu implementieren, verwenden Sie TLS. Weitere Informationen finden Sie unter „Verschlüsselte Sicherheitsprotokolle: TLS” auf Seite 17

2. Kanalauthentifizierungsdatensätze

   Siehe Pfeile 2 & 3. Die Authentifizierung kann unter Verwendung der IP-Adresse oder der TLS-definierten Namen auf der Sicherheitsstufe gesteuert werden. Eine Benutzer-ID kann auch blockiert werden, oder eine zugesicherte Benutzer-ID kann einer gültigen Benutzer-ID zugeordnet werden. Eine vollständige Beschreibung finden Sie in „Kanalauthentifizierungssätze” auf Seite 54.

3. Verbindungsauthentifizierung

   Siehe Pfeil 3. Der Client sendet eine ID und ein Kennwort. Weitere Informationen finden Sie unter „Verbindungsauthentifizierung: Konfiguration” auf Seite 76.

4. Kanalsicherheitsexits

   Siehe Pfeil 2. Die Kanalsicherheitsexits für Client-zu-Server-Kommunikation können auf die gleiche Weise funktionieren wie für Server-zu-Server-Kommunikation. Es kann ein protokollunabhängiges Paar von Exits geschrieben werden, um die gegenseitige Authentifizierung sowohl des Clients als auch des Servers zu ermöglichen. Eine vollständige Beschreibung finden Sie im Abschnitt Kanalsicherheitsexitprogramme .

5. Identifikation, die an einen Kanalsicherheitsexit übergeben wird

   Siehe Pfeil 3. In Client-zu-Server-Kommunikation müssen die Kanalsicherheitsexits nicht als Paar arbeiten. Der Exit auf IBM MQ-Clientseite kann weggelassen werden. In diesem Fall wird die Benutzer-ID in den Kanaldeskriptor (MQCD) gestellt, und der serverseitige Sicherheitsexit kann die Benutzer-ID ändern, falls erforderlich.

   IBM MQ MQI clients sendet auch zusätzliche Informationen, um die Identifikation zu unterstützen.

   - Die Benutzer-ID, die an den Server übergeben wird, ist die derzeit angemeldete Benutzer-ID auf dem Client.
   - Die Sicherheits-ID des derzeit angemeldeten Benutzers.

   Der Serversicherheitsexit kann mit den Werten der Benutzer-ID und, falls verfügbar, der Sicherheits-ID die Identität des IBM MQ MQI clients ermitteln.

Ab IBM MQ 8.0 können Sie Kennwörter senden, die in der MQCSP-Struktur enthalten sind.

Warnung: In einigen Fällen wird das Kennwort in einer MQCSP-Struktur für eine Clientanwendung über ein Netz in Klartext gesendet. Um sicherzustellen, dass die Kennwörter der Clientanwendung ordnungsgemäß geschützt sind, finden Sie weitere Informationen in „MQCSP-Kennwortschutz” auf Seite 34.

Benutzer-IDs

Wenn Sie Benutzer-IDs für Clientanwendungen erstellen, dürfen die Benutzer-IDs nicht länger als die maximal zulässige Länge sein. Sie dürfen die reservierten Benutzer-IDs UNKNOWN und NOBODY nicht verwenden. Wenn der Server, zu dem der Client eine Verbindung herstellt, ein IBM MQ for Windows-Server ist, müssen Sie die Verwendung des at-Zeichens @ mit Escape-Zeichen versehen. Die zulässige Länge von Benutzer-IDs ist abhängig von der Plattform, die für den Server verwendet wird:

- z/OS Linux AIX Unter z/OS, AIX and Linux beträgt die maximale Länge einer Benutzer-ID 12 Zeichen.
- IBM i Unter IBM i beträgt die maximale Länge einer Benutzer-ID 10 Zeichen.
- Windows Wenn sich unter Windowssowohl der IBM MQ MQI client als auch der IBM MQ -Server unter Windowsbefinden und der Server Zugriff auf die Domäne hat, in der die Client-Benutzer-ID definiert ist, beträgt die maximale Länge einer Benutzer-ID 20 Zeichen. Wenn es sich beim IBM MQ-Server allerdings nicht um einen Windows-Server handelt, wird die Benutzer-ID auf 12 Zeichen abgeschnitten.
- Wenn Sie die MQCSP-Struktur verwenden, um Berechtigungsnachweise zu übergeben, beträgt die maximale Länge einer Benutzer-ID 1024 Zeichen. Die MQCSP-Struktur-Benutzer-ID kann nicht verwendet werden, um die von IBM MQ für die Autorisierung verwendete maximale Benutzer-ID zu umgehen. Weitere Informationen zur MQCSP-Struktur finden Sie unter „Benutzer mit der MQCSP-Struktur identifizieren und authentifizieren” auf Seite 359.

Auf AIX and Linux-Systemen ist der Standardwert, dass Benutzer-IDs für die Authentifizierung verwendet werden, und Gruppen werden für die Autorisierung verwendet. Sie können diese Systeme jedoch so konfigurieren, dass sie für Benutzer-IDs autorisiert werden. Weitere Informationen finden Sie unter „Benutzerbasierte OAM-Berechtigungen unter AIX and Linux” auf Seite 377. Windows -Systeme können sowohl Benutzer-IDs für die Authentifizierung als auch für die Berechtigung und Gruppen für die Berechtigung verwenden.

Wenn Sie Service-Accounts erstellen, ohne auf Gruppen zu achten, und alle Benutzer-IDs unterschiedlich zu autorisieren, kann jeder Benutzer auf die Informationen jedes anderen Benutzers zugreifen.

Eingeschränkte Benutzer-IDs

Die Benutzer-IDs `UNKNOWN` und Gruppen `NOBODY` haben besondere Bedeutungen für IBM MQ. Die Erstellung einer Benutzer-ID im Betriebssystem `UNKNOWN` oder einer Gruppe mit dem Namen `NOBODY` kann zu unbeabsichtigten Ergebnissen führen.

Benutzer-ID beim Herstellen einer Verbindung zu einem IBM MQ for Windows-Server

Windows

Ein IBM MQ for Windows-Server unterstützt die Verbindung von IBM MQ MQI client nicht, wenn der Client mit einer Benutzer-ID ausgeführt wird, die das Zeichen @ enthält, wie zum Beispiel `abc@d`. Der Rückkehrcode für den Aufruf `MQCONN` auf dem Client lautet MQRC_NOT_AUTHORIZED.

Sie können die Benutzer-ID jedoch mit zwei @ Zeichen (z. B. `abc@@d`) angeben. Die Verwendung des `id@domain` -Formats ist das bevorzugte Verfahren, um sicherzustellen, dass die Benutzer-ID in der richtigen Domäne konsistent aufgelöst wird; daher `abc@@d@domain`.

Planungsberechtigung

Planen Sie, welche Benutzer Administratorberechtigung erhalten sollen, und planen Sie, wie die Benutzer von Anwendungen berechtigt werden, IBM MQ-Objekte ordnungsgemäß zu verwenden, einschließlich derer, die eine Verbindung von einem IBM MQ MQI client herstellen.

Einzelpersonen oder Anwendungen müssen Zugriffsberechtigungen erteilt werden, damit IBM MQ verwendet werden kann. Welche Zugriffsberechtigung sie benötigen, hängt von den Rollen, die sie ausführen, und den Tasks, die sie ausführen müssen, ab. Die Berechtigung in IBM MQ kann in zwei Hauptkategorien unterteilt werden:

- Berechtigung zum Ausführen von Verwaltungsoperationen
- Berechtigungen für Anwendung zur Verwendung von IBM MQ

Beide Operationsklassen werden von derselben Komponente gesteuert, und eine Einzelperson kann die Berechtigung zum Ausführen beider Kategorien von Operationen erteilen.

In den folgenden Abschnitten finden Sie weitere Informationen zu bestimmten Berechtigungsbereichen, die Sie berücksichtigen müssen:

Berechtigung zum Verwalten von IBM MQ

IBM MQ-Administratoren benötigen die Berechtigung zum Ausführen verschiedener Funktionen. Diese Berechtigung wird auf unterschiedliche Weise auf verschiedenen Plattformen abgerufen.

IBM MQ-Administrator benötigen die folgenden Berechtigungen:

- Ausgabe von Befehlen zum Verwalten von IBM MQ.
- Windows Linux Verwenden Sie den IBM MQ Explorer.
- z/OS Verwenden der Operationen und Steuerkonsolen unter z/OS.
- z/OS Verwenden des IBM MQ-Dienstprogramms CSQUTIL unter z/OS.
- z/OS Zugriff auf Warteschlangenmanagerdatasets unter z/OS.

Weitere Informationen finden Sie im entsprechenden Thema zu Ihrem Betriebssystem.

ALW *Berechtigung zum Verwalten von IBM MQ auf AIX, Linux, and Windows-Systemen*

Ein IBM MQ-Administrator ist ein Mitglied der Gruppe 'mqm'. Diese Gruppe verfügt über Zugriff auf alle IBM MQ-Ressourcen und kann IBM MQ-Steuerbefehle ausgeben. Ein Administrator kann anderen Benutzern bestimmte Berechtigungen erteilen.

Um ein IBM MQ-Administrator auf AIX, Linux, and Windows-Systemen zu sein, muss ein Benutzer Mitglied der *mqm-Gruppe* sein. Diese Gruppe wird bei der Installation von IBM MQ automatisch erstellt. Um Benutzern die Ausgabe von Steuerbefehlen zu ermöglichen, müssen Sie sie zur Gruppe 'mqm' hinzufügen. Dies schließt den Rootbenutzer unter AIX and Linux ein.

Benutzern, die nicht Mitglied der Gruppe 'mqm' sind, können Verwaltungsberechtigungen erteilt werden, aber sie können keine IBM MQ-Steuerbefehle ausgeben und sie sind nur zur Ausführung der Befehle berechtigt, für die ihnen Zugriff erteilt wurde.

Auf Windows -Systemen haben die Konten SYSTEM und Administrator uneingeschränkten Zugriff auf IBM MQ -Ressourcen.

Alle Mitglieder der Gruppe 'mqm' haben Zugriff auf alle IBM MQ-Ressourcen im System und können auch jeden Warteschlangenmanager verwalten, der im System ausgeführt wird. Dieser Zugriff kann nur widerrufen werden, wenn ein Benutzer aus der Gruppe 'mqm' entfernt wird. Auf Windows-Systemen haben Mitglieder der Administratorgruppe auch Zugriff auf alle IBM MQ-Ressourcen.

Administratoren können den Steuerbefehl **runmqsc** verwenden, um IBM MQ Script-Befehle (MQSC) auszugeben. Wenn **runmqsc** im indirekten Modus verwendet wird, um MQSC-Befehle an einen fernen Warteschlangenmanager zu senden, wird jeder MQSC-Befehl in einem Escape-PCF-Befehl eingebunden. Administratoren müssen über die erforderlichen Berechtigungen für die MQSC-Befehle verfügen, die vom fernen WS-Manager verarbeitet werden sollen.

Der IBM MQ Explorer gibt PCF-Befehle für die Ausführung von Verwaltungstasks aus. Administratoren benötigen keine weiteren Berechtigungen für die Verwendung des IBM MQ Explorer, um einen Warteschlan-

genmanager auf dem lokalen System verwalten zu können. Wenn ein Warteschlangenmanager auf einem anderen System vom IBM MQ Explorer verwaltet wird, müssen Administratoren über die erforderlichen Berechtigungen verfügen, damit die PCF-Befehle vom fernen Warteschlangenmanager verarbeitet werden können.

Weitere Informationen zu den Berechtigungsprüfungen, die bei der Verarbeitung von PCF-und MQSC-Befehlen durchgeführt werden, finden Sie in den folgenden Abschnitten:

- Informationen zu Befehlen für Warteschlangenmanager, Warteschlangen, Kanäle, Prozesse, Namenslisten und Authentifizierungsinformationsobjekte finden Sie in „Berechtigungen für Anwendung zur Verwendung von IBM MQ” auf Seite 98.
- Informationen zu Befehlen, die auf Kanälen, Kanalinitiatoren, Empfangsprogrammen und Clustern ausgeführt werden, finden Sie unter Kanalsicherheit .
- z/OS Informationen für MQSC-Befehle, die vom Befehlsserver unter IBM MQ for z/OS verarbeitet werden, finden Sie unter „Befehlssicherheit und Sicherheit der Befehlsressourcen unter z/OS” auf Seite 96.

Weitere Informationen zu der Berechtigung, die Sie für die Verwaltung von IBM MQ for AIX, Linux, and Windows-Systemen benötigen, finden Sie in den zugehörigen Informationen.

IBM i *Berechtigung für die Verwaltung von IBM MQ unter IBM i*

Als IBM MQ-Administrator unter IBM i müssen Sie ein Mitglied der *Gruppe QMQMADM* sein. Diese Gruppe verfügt über Eigenschaften, die denen der Gruppe 'mqm' auf AIX, Linux, and Windows-Systemen ähneln. Insbesondere wird die Gruppe QMQMADM bei der Installation von IBM MQ for IBM i erstellt und die Mitglieder der Gruppe QMQMADM haben Zugriff auf alle IBM MQ-Ressourcen im System. Sie haben auch Zugriff auf alle IBM MQ-Ressourcen, wenn Sie die Berechtigung *ALLOBJ haben.

Administratoren können CL-Befehle verwenden, um IBM MQ zu verwalten.Einer dieser Befehle ist GRTMQMAUT, der für die Erteilung von Berechtigungen für andere Benutzer verwendet wird. Ein anderer Befehl, STRMQMMQSC, ermöglicht es einem Administrator, MQSC-Befehle an einen lokalen WS-Manager auszugeben.

Es gibt zwei Gruppen von CL-Befehlen, die von IBM MQ for IBM i bereitgestellt werden:

Gruppe 1
Um einen Befehl in dieser Kategorie absetzen zu können, muss ein Benutzer Mitglied der Gruppe QMQMADM sein oder die Berechtigung *ALLOBJ besitzen. GRTMQMAUT und STRMQMMQSC gehören zum Beispiel zu dieser Kategorie.

Gruppe 2
Um einen Befehl in dieser Kategorie absetzen zu können, muss ein Benutzer nicht Mitglied der Gruppe QMQMADM sein oder die Berechtigung *ALLOBJ besitzen. Stattdessen sind zwei Berechtigungsstufen erforderlich:

- Der Benutzer benötigt die IBM i-Berechtigung für die Verwendung des Befehls. Diese Berechtigung wird mit dem Befehl GRTOBJAUT erteilt.
- Der Benutzer benötigt die IBM MQ-Berechtigung für die Zugriff auf ein IBM MQ-Objekt, das dem Befehl zugeordnet ist. Diese Berechtigung wird mit dem Befehl GRTMQMAUT erteilt.

Die folgenden Beispiele zeigen Befehle in dieser Gruppe:

- CRTMQMQ, MQM-Warteschlange erstellen
- CHGMQMPRC, MQM-Prozess ändern
- DLTMQMNL, MQM-Namensliste löschen
- DSPMQMAUTI, MQM-Authentifizierungsinformationen anzeigen
- CRTMQMCHL, MQM-Kanal erstellen

Weitere Informationen zu dieser Gruppe von Befehlen finden Sie in „Berechtigungen für Anwendung zur Verwendung von IBM MQ” auf Seite 98.

Eine vollständige Liste der Befehle der Gruppe 1 und 2 finden Sie unter „Zugriffsberechtigungen für IBM MQ-Objekte unter IBM i” auf Seite 170

Weitere Informationen zu der Berechtigung, die Sie für die Verwaltung von IBM MQ unter IBM ibenötigen, finden Sie im Abschnitt Verwalten von IBM i .

z/OS *Berechtigung für die Verwaltung von IBM MQ unter z/OS*

In dieser Themensammlung werden die verschiedenen Aspekte der Berechtigung beschrieben, die Sie für die Verwaltung von IBM MQ for z/OS benötigen.

z/OS *Berechtigungsprüfungen unter z/OS*

IBM MQ for z/OS verwendet die System Authorization Facility (SAF), um Anforderungen für Berechtigungsprüfungen an einen externen Sicherheitsmanager (ESM), wie z. B. die z/OS Security Server Resource Access Control Facility (RACF), weiterzuleiten. IBM MQ überprüft keine eigenen Berechtigungsprüfungen.

Es wird vorausgesetzt, dass Sie RACF als ESM verwenden. Wenn Sie einen anderen ESM verwenden, müssen Sie die für RACF bereitgestellten Informationen so interpretieren, dass sie für Ihren ESM relevant sind.

Sie können angeben, ob Berechtigungsprüfungen für jeden Warteschlangenmanager einzeln oder für jeden Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange aktiviert oder inaktiviert werden sollen.Diese Stufe der Steuerung wird als *Subsystemsicherheit* bezeichnet. Wenn Sie die Subsystemsicherheit für einen bestimmten Warteschlangenmanager inaktivieren, werden keine Berechtigungsprüfungen für diesen Warteschlangenmanager durchgeführt.

Wenn Sie die Subsystemsicherheit für einen bestimmten Warteschlangenmanager aktivieren, können Berechtigungsprüfungen auf zwei Ebenen ausgeführt werden:

Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange
: Berechtigungsprüfungen verwenden RACF-Profile, die von allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange gemeinsam genutzt werden. Dies bedeutet, dass es weniger Profile gibt, die definiert und verwaltet werden, wodurch die Sicherheitsverwaltung vereinfacht wird.

Sicherheit auf WS-Managerebene
: In Berechtigungsprüfungen werden spezielle RACF-Profile für den Warteschlangenmanager verwendet.

Sie können eine Kombination aus der Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange und des Warteschlangenmanagers verwenden. Sie können in Profilen beispielsweise speziell für Warteschlangenmanager angeben, dass die Gruppe mit gemeinsamer Warteschlange überschrieben wird, der sie zugeordnet sind.

Die Sicherheit auf der Ebene des Subsystems, die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange und die Sicherheit auf Warteschlangenmanagerebene werden durch das Definieren von *Schalterprofilen* aktiviert oder inaktiviert. Ein Schalterprofil ist ein übliches RACF-Profil, das für IBM MQ eine besondere Bedeutung hat.

z/OS *Befehlssicherheit und Sicherheit der Befehlsressourcen unter z/OS*

Die Befehlssicherheit bezieht sich auf die Berechtigung zur Ausgabe eines Befehls; die Befehlsressourcenberechtigung bezieht sich auf die Berechtigung zum Ausführen einer Operation für eine Ressource. Beide werden mithilfe von RACF-Klassen implementiert.

Berechtigungsprüfungen werden ausgeführt, wenn ein IBM MQ-Administrator einen MQSC-Befehl ausgibt. Dies wird als *Befehlssicherheit* bezeichnet.

Zur Implementierung der Befehlssicherheit müssen Sie bestimmte RACF-Profile definieren und den erforderlichen Gruppen und Benutzer-IDs Zugriff auf diese Profile auf den erforderlichen Ebenen erteilen. Der Name eines Profils für die Befehlssicherheit enthält den Namen eines MQSC-Befehls.

Einige MQSC-Befehle führen eine Operation auf einer IBM MQ-Ressource aus, wie beispielsweise der Befehl DEFINE QLOCAL, mit dem eine lokale Warteschlange erstellt wird. Wenn ein Administrator einen

MQSC-Befehl ausgibt, werden Berechtigungsprüfungen durchgeführt, um festzustellen, ob die angeforderte Operation für die im Befehl angegebene Ressource ausgeführt werden kann. Dies wird als *Befehlsressourcensicherheit* bezeichnet.

Zur Implementierung der Sicherheit für die Befehlsressourcen müssen Sie bestimmte RACF-Profile definieren und den erforderlichen Gruppen und Benutzer-IDs Zugriff auf diese Profile auf den erforderlichen Ebenen erteilen. Der Name eines Profils für die Befehlsressourcensicherheit enthält den Namen einer IBM MQ-Ressource und den zugehörigen Typ (QUEUE, PROCESS, NAMELIST, TOPIC, AUTHINFO oder CHANNEL).

Die Sicherheit der Befehlssicherheit und die Sicherheit der Befehlsressourcen sind unabhängig. Wenn ein Administrator z. B. den folgenden Befehl ausgibt:

```
DEFINE QLOCAL(MOON.EUROPA)
```

werden die folgenden Berechtigungsprüfungen durchgeführt:

- Die Befehlssicherheit überprüft, ob der Administrator berechtigt ist, den Befehl DEFINE QLOCAL auszugeben.
- Die Befehlsressourcensicherheit überprüft, ob der Administrator berechtigt ist, eine Operation in der lokalen Warteschlange mit dem Namen MOON.EUROPA auszuführen.

Die Sicherheit der Befehlssicherheit und die Sicherheit der Befehlsressourcen können durch Definieren von Schalterprofilen aktiviert oder inaktiviert werden.

z/OS *MQSC-Befehle und Eingabewarteschlange für Systembefehle unter z/OS*

In diesem Abschnitt wird erläutert, wie der Befehlsserver MQSC-Befehle verarbeitet, die an die Eingabewarteschlange für Systembefehle unter z/OS übertragen werden.

Die Befehlssicherheit und die Sicherheit der Befehlsressourcen werden auch verwendet, wenn der Befehlsserver eine Nachricht abruft, die einen MQSC-Befehl aus der Eingabewarteschlange des Systembefehls enthält.Die Benutzer-ID, die für die Berechtigungsprüfungen verwendet wird, ist die im Feld *Benutzer-ID* im Nachrichtendeskriptor der Nachricht, die den MQSC-Befehl enthält, gefunden.Die Benutzer-ID, die für die Berechtigungsprüfungen Diese Benutzer-ID muss über die erforderlichen Berechtigungen auf dem Warteschlangenmanager verfügen, auf dem der Befehl verarbeitet wird. Weitere Informationen zum Feld *UserIdentifier* und zu seiner Definition finden Sie im Abschnitt Nachrichtenkontext .

Nachrichten, die MQSC-Befehle enthalten, werden unter den folgenden Umständen an die Eingabewarteschlange des Systembefehls gesendet:

- Die Operationen und Steuerkonsolen senden MQSC-Befehle an die Eingabewarteschlange des Systembefehls des Zielwarteschlangenmanagers. Die MQSC-Befehle entsprechen den Aktionen, die Sie in den Anzeigen auswählen. Das Feld *UserIdentifier* in jeder Nachricht wird auf die TSO-Benutzer-ID des Administrators gesetzt.
- Mit der COMMAND-Funktion des IBM MQ-Dienstprogramms CSQUTIL werden die MQSC-Befehle im Eingabedataset an die Eingabewarteschlange für Systembefehle des Zielwarteschlangenmanagers gesendet. Die Funktionen COPY und EMPTY senden die Befehle DISPLAY QUEUE und DISPLAY STGCLASS. Das Feld *UserIdentifier* in jeder Nachricht wird auf die Jobbenutzer-ID gesetzt.
- Die MQSC-Befehle in den CSQINPX-Dateien werden an die Befehlseingabewarteschlange des Systembefehls des Warteschlangenmanagers gesendet, mit dem der Kanalinitiator verbunden ist. Das Feld *UserIdentifier* in jeder Nachricht wird auf die Benutzer-ID des Kanalinitiatoradressraums gesetzt.

  Es werden keine Berechtigungsprüfungen durchgeführt, wenn MQSC-Befehle von den Datensätzen CSQINP1 und CSQINP2 ausgegeben werden. Sie können über den RACF-Datenschutz festlegen, wer diese Datasets aktualisieren kann.
- Innerhalb einer Gruppe mit gemeinsamer Warteschlange kann ein Kanalinitiator Befehle des Typs START CHANNEL an die Eingabewarteschlange für Systembefehle des Warteschlangenmanagers senden, mit dem sie verbunden ist. Ein Befehl wird gesendet, wenn ein abgehender Kanal, der eine gemeinsam genutzte Übertragungswarteschlange verwendet, durch Triggern gestartet wird. Das Feld *UserIdentifier* in jeder Nachricht wird auf die Benutzer-ID des Kanalinitiatoradressraums gesetzt.

- Eine Anwendung kann MQSC-Befehle an eine Eingabewarteschlange des Systembefehls senden. Standardmäßig wird das Feld *UserIdentifier* in jeder Nachricht auf die Benutzer-ID gesetzt, die der Anwendung zugeordnet ist.
- Auf Systemen mit AIX, Linux, and Windows kann der Steuerbefehl **runmqsc** im indirekten Modus verwendet werden, um MQSC-Befehle an die Eingabewarteschlange für Systembefehle eines Warteschlangenmanagers unter z/OS zu senden. Das Feld *UserIdentifier* in jeder Nachricht wird auf die Benutzer-ID des Administrators gesetzt, der den Befehl **runmqsc** ausgegeben hat.

z/OS *Zugriff auf Datensätze von Warteschlangenmanagern unter z/OS*

IBM MQ for z/OS-Administratoren benötigen die Berechtigung für den Zugriff auf die Datensätze von Warteschlangenmanagern. In diesem Abschnitt wird erläutert, welche Datensätze einen RACF-Schutz benötigen.

Zu diesen Datensätzen gehören:

- Die Datensätze, auf die von CSQINP1, CSQINP2 und CSQINPT in der gestarteten Taskprozedur des Warteschlangenmanagers Bezug genommen wird.
- Die Seitengruppen des Warteschlangenmanagers, aktive Protokolldateien, Archivprotokolldateien und Bootstrap-Dateigruppen (BSDSs)
- Die Datensätze, auf die CSQXLIB und CSQINPX in der gestarteten Taskprozedur des Kanalinitiators Bezug genommen haben

Sie müssen die Dateien schützen, damit kein nicht berechtigter Benutzer einen WS-Manager starten oder Zugriff auf alle WS-Manager-Daten erhalten kann. Verwenden Sie dazu den Schutz für die RACF-Datensätze.

Berechtigungen für Anwendung zur Verwendung von IBM MQ

Wenn Anwendungen auf Objekte zugreifen, benötigen die Benutzer-IDs, die den Anwendungen zugeordnet sind, die entsprechende Berechtigung.

Anwendungen können durch die Ausgabe von MQI-Aufrufen auf die folgenden IBM MQ-Objekte zugreifen:

- Warteschlangenmanager
- Warteschlangen
- Prozesse
- Namenslisten
- Themen

Anwendungen können auch PCF-Befehle verwenden, um IBM MQ-Objekte zu verwalten. Wenn der PCF-Befehl verarbeitet wird, verwendet er den Berechtigungskontext der Benutzer-ID, die die PCF-Nachricht eingibt.

Zu diesen Anwendungen gehören in diesem Kontext Anwendungen, die von Benutzern und anderen Herstellen geschrieben wurden sowie die mit IBM MQ for z/OS bereitgestellten Anwendungen. Zu den mit IBM MQ for z/OS bereitgestellten Anwendungen gehören:

- Die Operationen und Steuerkonsolen
- Das IBM MQ-Dienstprogramm CSQUTIL
- Das Dienstprogramm für die Warteschlange für nicht zustellbare Nachrichten, CSQUDLQH,

Anwendung, die IBM MQ classes for Java, IBM MQ classes for JMS, IBM MQ classes for .NET oder die Message Service Clients for C/C++ und .NET verwenden, verwenden die MQI indirekt.

Nachrichtenkanalagenten geben ebenfalls MQI-Aufrufe aus; daher benötigen die den Nachrichtenkanalagenten zugeordneten Benutzer-IDs eine Zugriffsberechtigung für diese IBM MQ-Objekte. Weitere Informationen zu diesen Benutzer-IDs und den erforderlichen Berechtigungen finden Sie in „Kanalberechtigung” auf Seite 122.

Unter z/OS können Anwendungen auch MQSC-Befehle für den Zugriff auf diese IBM MQ-Objekte verwenden; in diesem Fall übernehmen allerdings die Befehlssicherheit und die Sicherheit der Befehlsressourcen die Berechtigungsprüfungen. z/OS Weitere Informationen finden Sie unter „Befehlssicherheit und Sicherheit der Befehlsressourcen unter z/OS" auf Seite 96 und „MQSC-Befehle und Eingabewarteschlange für Systembefehle unter z/OS" auf Seite 97.

Unter IBM i benötigt ein Benutzer, der einen CL-Befehl in Gruppe 2 ausgibt, möglicherweise die Zugriffsberechtigung auf ein IBM MQ-Objekt, das dem Befehl zugeordnet ist. Weitere Informationen finden Sie in „Wenn Berechtigungsprüfungen durchgeführt werden" auf Seite 99.

Wenn Berechtigungsprüfungen durchgeführt werden

Berechtigungsprüfungen werden durchgeführt, wenn eine Anwendung versucht, auf einen WS-Manager, eine Warteschlange, einen Prozess oder eine Namensliste zuzugreifen.

Unter IBM i können Berechtigungsprüfungen auch dann ausgeführt werden, wenn ein Benutzer einen CL-Befehl in Gruppe 2 ausgibt, die auf eines der IBM MQ-Objekte zugreift. Die Prüfungen werden unter den folgenden Umständen ausgeführt:

Wenn eine Anwendung über einen MQCONN -oder MQCONNX -Aufruf eine Verbindung zu einem Warteschlangenmanager herstellt

Der Warteschlangenmanager fragt das Betriebssystem nach der Benutzer-ID, die der Anwendung zugeordnet ist. Der Warteschlangenmanager prüft dann, ob die Benutzer-ID berechtigt ist, eine Verbindung zu dieser herzustellen, und behält die Benutzer-ID für zukünftige Prüfungen bei.

Benutzer müssen sich sich bei IBM MQ anmelden. IBM MQ setzt voraus, dass sich die Benutzer am zugrunde liegenden Betriebssystem angemeldet haben und dort authentifiziert sind.

Wenn eine Anwendung ein IBM MQ-Objekt mit einem MQOPEN- oder MQPUT1-Aufruf öffnet

Alle Berechtigungsprüfungen werden ausgeführt, wenn ein Objekt geöffnet wird, nicht wenn später auf das Objekt zugegriffen wird. Berechtigungsprüfungen werden z. B. ausgeführt, wenn eine Anwendung eine Warteschlange öffnet. Sie werden nicht ausgeführt, wenn die Anwendung Nachrichten in die Warteschlange einreiht oder Nachrichten aus der Warteschlange abruft.

Wenn eine Anwendung ein Objekt öffnet, gibt sie die Typen der Operation an, die sie für das Objekt ausführen muss. Eine Anwendung kann z. B. eine Warteschlange öffnen, um die Nachrichten in ihr zu durchsuchen, Nachrichten von ihr abzurufen, aber keine Nachrichten in sie zu stellen. Für jeden Typ von Operation prüft der Warteschlangenmanager, ob die der Anwendung zugeordnete Benutzer-ID die Berechtigung zum Ausführen dieser Operation hat.

Wenn eine Anwendung eine Warteschlange öffnet, werden die Berechtigungsprüfungen für das Objekt ausgeführt, das im Feld `ObjectName` des Objektdeskriptors angegeben ist. Das Feld `ObjectName` wird in den Aufrufen `MQOPEN` oder `MQPUT1` verwendet. Wenn es sich bei dem Objekt um eine Aliaswarteschlange oder eine Definition einer fernen Warteschlange handelt, werden die Berechtigungsprüfungen für das Objekt selbst durchgeführt. Sie werden nicht in der Warteschlange ausgeführt, in die die Aliaswarteschlange oder die Definition der fernen Warteschlange aufgelöst wird. Dies bedeutet, dass der Benutzer keine Berechtigung zum Zugriff auf ihn benötigt. Begrenzen Sie die Berechtigung zum Erstellen von Warteschlangen für privilegierte Benutzer. Wenn Sie dies nicht tun, können Benutzer die normale Zugriffssteuerung umgehen, indem Sie einfach einen Aliasnamen erstellen.

Eine Anwendung kann explizit auf eine ferne Warteschlange verweisen. Sie setzt die Felder `ObjectName` und `ObjectQMgrName` in dem Objektdeskriptor auf die Namen der fernen Warteschlange und des fernen Warteschlangenmanagers. Die Berechtigungsprüfungen werden für die Übertragungswarteschlange mit demselben Namen wie der ferne Warteschlangenmanager ausgeführt:

- z/OS Unter z/OSwird das RACF -Warteschlangenprofil überprüft, das mit dem Namen des fernen Warteschlangenmanagers übereinstimmt, und es wird geprüft, ob diese Übertragungswarteschlange lokal definiert ist.
- Multi Unter Multiplatformswird das RQMNAME-Profil überprüft, das mit dem Namen des fernen Warteschlangenmanagers übereinstimmt, wenn Clustering verwendet wird.

Eine Anwendung kann explizit auf eine Clusterwarteschlange verweisen, indem Sie das Feld `ObjectName` im Objektdeskriptor auf den Namen der Clusterwarteschlange setzen. Die Berechtigungsprüfungen werden für die Clusterübertragungswarteschlange `SYSTEM.CLUSTER.TRANSMIT.QUEUE` ausgeführt.

Die Berechtigung für eine dynamische Warteschlange basiert auf der Modellwarteschlange, aus der sie abgeleitet wird, ist aber nicht unbedingt identisch; siehe Anmerkung 1 .

Die Benutzer-ID, die der Queue Manager für die Berechtigungsprüfungen verwendet, wird über das Betriebssystem abgerufen. Die Benutzer-ID wird abgerufen, wenn die Anwendung eine Verbindung zum WS-Manager herstellt. Eine entsprechend berechtigte Anwendung kann einen `MQOPEN` -Aufruf ausgeben, der eine alternative Benutzer-ID angibt. Anschließend werden Zugriffssteuerungsprüfungen für die alternative Benutzer-ID durchgeführt. Bei Verwendung einer alternativen Benutzer-ID wird die der Anwendung zugeordnete Benutzer-ID nicht geändert, sondern nur die Benutzer-ID, die für den Zugriff auf Steuerprüfungen verwendet wird.

Wenn eine Anwendung ein Thema mit einem `MQSUB` -Aufruf subskribiert.

Wenn eine Anwendung ein Thema subskribiert, gibt sie die Art der Operation an, die sie ausführen muss. Es wird entweder eine Subskription erstellt, eine vorhandene Subskription geändert oder eine vorhandene Subskription wieder aufgenommen, ohne sie zu ändern. Für jeden Typ von Operation prüft der Warteschlangenmanager, ob die Benutzer-ID, die der Anwendung zugeordnet ist, über die Berechtigung zum Ausführen der Operation verfügt.

Wenn eine Anwendung ein Thema subskribiert, werden die Berechtigungsprüfungen für Themenobjekte durchgeführt, die in der Themenstruktur gefunden werden. Die Themenobjekte befinden sich in oder oberhalb des Punktes in der Themenstruktur, in der die Anwendung subskribiert hat. Bei den Berechtigungsprüfungen kann es sich um Prüfungen auf mehr als ein Themenobjekt handeln. Die Benutzer-ID, die der Queue Manager für die Berechtigungsprüfungen verwendet, wird über das Betriebssystem abgerufen. Die Benutzer-ID wird abgerufen, wenn die Anwendung eine Verbindung zum WS-Manager herstellt.

Der Warteschlangenmanager führt Berechtigungsprüfungen für Subskribentenwarteschlangen aus, jedoch nicht in den verwalteten Warteschlangen.

Wenn eine Anwendung eine permanente dynamische Warteschlange mit einem `MQCLOSE` -Aufruf löscht

Die im Aufruf `MQCLOSE` angegebene Objektkennung ist nicht unbedingt dieselbe, die vom Aufruf `MQOPEN` zurückgegeben wird, der die permanente dynamische Warteschlange erstellt hat. Ist dies der Fall, überprüft der Warteschlangenmanager die Benutzer-ID, die der Anwendung zugeordnet ist, die den Aufruf `MQCLOSE` ausgegeben hat. Es prüft, ob die Benutzer-ID berechtigt ist, die Warteschlange zu löschen.

Wenn eine Anwendung, die eine Subskription schließt, um sie zu entfernen, nicht erstellt wurde, ist die entsprechende Berechtigung erforderlich, um sie zu entfernen.

Wenn ein PCF-Befehl für ein IBM MQ-Objekt vom Befehlsserver verarbeitet wird

Diese Regel schließt den Fall ein, in dem ein PCF-Befehl auf einem Authentifizierungsinformationsobjekt ausgeführt wird.

Die Benutzer-ID, die für die Berechtigungsprüfungen verwendet wird, wird im Feld `UserIdentifier` im Nachrichtendeskriptor des PCF-Befehls angezeigt. Diese Benutzer-ID muss über die erforderlichen Berechtigungen auf dem Warteschlangenmanager verfügen, auf dem der Befehl verarbeitet wird. Der entsprechende MQSC-Befehl, der in einem Escape-PCF-Befehl eingebunden ist, wird auf die gleiche Weise behandelt. Weitere Informationen zum Feld `UserIdentifier` und zu seiner Definition finden Sie in „Nachrichtenkontext” auf Seite 101.

IBM i **Wenn ein Benutzer unter IBM i einen CL-Befehl in Gruppe 2 ausgibt, mit dem ein IBM MQ-Objekt bearbeitet wird**

Diese Regel schließt den Fall ein, in dem ein CL-Befehl in Gruppe 2 auf einem Authentifizierungsinformationsobjekt ausgeführt wird.

Mit den Prüfungen wird ermittelt, ob der Benutzer zur Bearbeitung eines IBM MQ-Objekts berechtigt ist, das dem Befehl zugeordnet ist. Die Prüfungen werden ausgeführt, es sei denn, der Benutzer ist

Mitglied der Gruppe QMQMADM oder verfügt über die Berechtigung *ALLOBJ . Die erforderliche Berechtigung richtet sich nach dem Typ der Operation, die der Befehl für das Objekt ausführt. Beispiel: Der Befehl **CHGMQMQ**, Change MQM Queue, erfordert die Berechtigung, die Attribute der durch den Befehl angegebenen Warteschlange zu ändern. Im Gegensatz dazu benötigt der Befehl **DSPMQMQ**, Display MQM Queue, die Berechtigung zum Anzeigen der Attribute der mit dem Befehl angegebenen Warteschlange.

Viele Befehle arbeiten auf mehr als einem Objekt. Zur Ausgabe des Befehls **DLTMQMQ**, Delete MQM Queue, sind beispielsweise die folgenden Berechtigungen erforderlich:

- Die Berechtigung zum Herstellen einer Verbindung zu dem durch den Befehl angegebenen Warteschlangenmanager.
- Die Berechtigung zum Löschen der Warteschlange, die durch den Befehl angegeben wurde.

Einige Befehle arbeiten überhaupt nicht an Objekt. In diesem Fall benötigt der Benutzer nur die Berechtigung IBM i , um einen dieser Befehle auszugeben. **STRMQMLSR** Start MQM Listener, ist ein Beispiel für einen solchen Befehl.

Alternative Benutzerberechtigung

Wenn eine Anwendung ein Objekt öffnet oder ein Thema subskribiert, kann die Anwendung eine Benutzer-ID im MQOPEN-, MQPUT1-oder MQSUB-Aufruf angeben. Er kann den WS-Manager bitten, diese Benutzer-ID für Berechtigungsprüfungen zu verwenden, anstatt die der Anwendung zugeordnete zu verwenden.

Die Anwendung kann das Objekt nur öffnen, wenn die beiden folgenden Bedingungen erfüllt sind:

- Die Benutzer-ID, die der Anwendung zugeordnet ist, verfügt über die Berechtigung, eine andere Benutzer-ID für Berechtigungsprüfungen zu liefern. Die Anwendung hat die Berechtigung *alternative Benutzerberechtigung* .
- Die von der Anwendung bereitgestellte Benutzer-ID verfügt über die Berechtigung zum Öffnen des Objekts für die angeforderten Typen von Operationen oder zum Subskribieren des Themas.

Nachrichtenkontext

Nachrichtenkontext ermöglicht es der Anwendung, die eine Nachricht abruft, um Informationen über den Absender der Nachricht zu erhalten. Die betreffenden Informationen befinden sich in den Feldern des Nachrichtendeskriptors, die in drei logische Bereiche eingeteilt sind.

Diese Teile sind wie folgt:

Identitätskontext
Diese Felder enthalten Informationen über den Benutzer der Anwendung, die die Nachricht in die Warteschlange gestellt hat.

Ursprungskontext
Diese Felder enthalten Informationen über die Anwendung selbst sowie den Zeitpunkt, zu dem die Nachricht eingereiht wurde.

Benutzerkontext
Diese Felder enthalten Nachrichteneigenschaften, die Anwendungen verwenden können, um Nachrichten auszuwählen, die vom WS-Manager geliefert werden sollen.

Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, kann die Anwendung den WS-Manager auffordern, die Kontextinformationen in der Nachricht zu generieren. Dies ist die Standardaktion. Alternativ kann auch angegeben werden, dass die Kontextfelder keine Informationen enthalten sollen. Die Benutzer-ID, die einer Anwendung zugeordnet ist, benötigt keine Sonderberechtigung, um eine dieser beiden Anwendungen zu machen.

Eine Anwendung kann die Identitätskontextfelder in einer Nachricht festlegen, so dass der Warteschlangenmanager den Ursprungskontext generieren kann, oder er kann alle Kontextfelder festlegen. Eine Anwendung kann auch die Identitätskontextfelder aus einer Nachricht, die sie abgerufen hat, an eine Nachricht übergeben, die sie in eine Warteschlange eingibt, oder sie kann alle Kontextfelder übergeben. Die Benutzer-ID, die einer Anwendung zugeordnet ist, erfordert jedoch die Berechtigung zum Festlegen oder Übergeben von Kontextinformationen. Eine Anwendung gibt an, dass sie Kontextinformationen fest-

legen oder übergeben will, wenn sie die Warteschlange öffnet, in der sie Nachrichten einlegen soll, und ihre Berechtigung wird zu diesem Zeitpunkt geprüft.

Im Folgenden finden Sie eine kurze Beschreibung der einzelnen Kontextfelder:

Identitätskontext

UserIdentifier
Die Benutzer-ID, die der Anwendung zugeordnet ist, die die Nachricht eingibt. Wenn der Warteschlangenmanager dieses Feld festlegt, wird er auf die Benutzer-ID gesetzt, die vom Betriebssystem abgerufen wird, wenn die Anwendung eine Verbindung zum Warteschlangenmanager herstellt.

AccountingToken
Informationen, die verwendet werden können, um die Arbeit zu berechnen, die als Ergebnis der Nachricht ausgeführt wurde.

ApplIdentityData
Wenn die Benutzer-ID, die einer Anwendung zugeordnet ist, die Berechtigung zum Festlegen der Identitätskontextfelder oder zum Festlegen aller Kontextfelder hat, kann die Anwendung dieses Feld auf einen beliebigen Wert im Zusammenhang mit der Identität setzen. Wenn der WS-Manager dieses Feld definiert, wird er auf Leerzeichen gesetzt.

Ursprungskontext

PutApplType
Die Art der Anwendung, von der die Nachricht eingereiht wurde, z. B. eine CICS-Transaktion.

PutApplName
Der Name der Anwendung, von der die Nachricht eingereiht wurde.

PutDate
Das Datum, an dem die Nachricht gestellt wurde.

PutTime
Die Uhrzeit, zu der die Nachricht gestellt wurde.

ApplOriginData
Wenn die Benutzer-ID, die einer Anwendung zugeordnet ist, die Berechtigung zum Festlegen aller Kontextfelder hat, kann die Anwendung dieses Feld auf einen beliebigen Wert im Zusammenhang mit dem Ursprung setzen. Wenn der WS-Manager dieses Feld definiert, wird er auf Leerzeichen gesetzt.

Benutzerkontext

Die folgenden Werte werden für **MQINQMP** oder **MQSETMP**unterstützt:

MQPD_USER_CONTEXT

Die Eigenschaft wird dem Benutzerkontext zugeordnet.

Um eine dem Benutzerkontext zugeordnete Eigenschaft über den MQSETMP-Aufruf festzulegen, ist keine besondere Berechtigung erforderlich.

Auf einem V7.0-oder einem nachfolgenden Warteschlangenmanager wird eine dem Benutzerkontext zugeordnete Eigenschaft gespeichert, wie für MQOO_SAVE_ALL_CONTEXT beschrieben. Ein MQPUT-Aufruf mit MQOO_PASS_ALL_CONTEXT bewirkt, dass die Eigenschaft aus dem gespeicherten Kontext in die neue Nachricht kopiert wird.

MQPD_NO_CONTEXT

Die Eigenschaft ist keinem Nachrichtenkontext zugeordnet.

Ein nicht erkannter Wert wird mit MQRC_PD_ERROR zurückgewiesen. Der Anfangswert dieses Felds lautet **MQPD_NO_CONTEXT**.

Eine detaillierte Beschreibung der einzelnen Kontextfelder finden Sie im Abschnitt MQMD-Nachrichtendeskriptor . Weitere Informationen zur Verwendung des Nachrichtenkontextes finden Sie im Abschnitt Nachrichtenkontext .

IBM i ALW *Berechtigung zum Arbeiten mit IBM MQ-Objekten auf IBM i AIX, Linux, and Windows-Systemen*

Die mit IBM MQ bereitgestellte Berechtigungsservicekomponente wird als *Objektberechtigungsmanager* (Object Authority Manager, OAM) bezeichnet. Sie ermöglicht die Zugriffssteuerung über Authentifizierungs-und Berechtigungsprüfungen.

Authentifizierung.

Die Authentifizierungsprüfung, die von dem mit IBM MQ bereitgestellten OAM durchgeführt wird, ist eine Basisauthentifizierung und wird nur in bestimmten Fällen ausgeführt. Es ist nicht beabsichtigt, die strengen Anforderungen zu erfüllen, die in einer hochsicheren Umgebung erwartet werden.

Das OAM führt seine Authentifizierungsprüfung durch, wenn eine Anwendung eine Verbindung zu einem Queue Manager herstellt, und die folgenden Bedingungen sind wahr:

- Wenn eine MQCSP-Struktur von der Verbindungsanwendung bereitgestellt wurde, und
- Für das Attribut *AuthenticationType* in der MQCSP-Struktur der Wert MQCSP_AUTH_USER_ID_AND_PWD angegeben wird.
- Der Wert CHCKLOCL oder CHKCCLNT auf dem konfigurierten AUTHINFO-Objekt ist nicht 'NONE'.

Die Authentifizierungsschritte im OAM überprüfen das Kennwort mit Hilfe von Betriebssystemservices, die möglicherweise für zusätzliche Prüfungen konfiguriert wurden, z. B., dass der Benutzername nicht zu viele falsche Kennwortprüfversuche hatte.

Es ist möglich, alternative Authentifizierungsverfahren zu verwenden, wenn Sie eine neue Berechtigungsservicekomponente schreiben oder einen von einem Anbieter beziehen.

Autorisierung.

Die Berechtigungsprüfungen sind umfassend und sollen die meisten normalen Anforderungen erfüllen.

Berechtigungsprüfungen werden ausgeführt, wenn eine Anwendung einen MQI-Aufruf ausgibt, um auf einen Warteschlangenmanager, eine Warteschlange, einen Prozess, ein Thema oder eine Namensliste zuzugreifen. Sie werden auch zu anderen Zeitpunkten ausgeführt, z. B., wenn ein Befehl vom Befehlsserver ausgeführt wird.

Auf IBM i IBM i-, AIX, Linux, and Windows-Systemen gibt der *Berechtigungsservice* über die Zugriffssteuerung an, wann ein MQI-Aufruf für den Zugriff auf ein IBM MQ-Objekt ausgegeben wird, bei dem es sich um einen Warteschlangenmanager, einen Prozess, ein Thema oder eine Namensliste handelt. Dazu gehören Prüfungen auf alternative Benutzerberechtigung und die Berechtigung zum Festlegen oder Übergeben von Kontextinformationen.

Windows Unter Windows erteilt der OAM den Mitgliedern der Administratorgruppe die Berechtigung, auf alle IBM MQ-Objekte zuzugreifen, selbst wenn UAC aktiviert ist. Außerdem hat das Konto SYSTEM auf Windows -Systemen uneingeschränkten Zugriff auf IBM MQ -Ressourcen.

Der Berechtigungsservice stellt zusätzlich Berechtigungsprüfungen bereit, wenn ein PCF-Befehl eines dieser IBM MQ-Objekte oder ein Authentifizierungsdatenobjekt ausführt. Der entsprechende MQSC-Befehl, der in einem Escape-PCF-Befehl eingebunden ist, wird auf die gleiche Weise behandelt.

IBM i Wenn der Benutzer kein Mitglieder der Gruppe QMQMADM ist oder die Berechtigung *ALLOBJ hat, stellt der Berechtigungsservice unter IBM i außerdem Berechtigungsprüfungen bereit, wenn ein Benutzer einen CL-Befehl in Gruppe 2 ausgibt, die auf einem dieser IBM MQ-Objekte oder einem Authentifizierungsdatenobjekt ausgeführt wird.

Der Berechtigungsservice ist ein *installierbarer Service* , d. er bedeutet, dass er von einer oder mehreren *installierbaren Servicekomponenten* implementiert wird. Jede Komponente wird über eine dokumentierte Schnittstelle aufgerufen. Dadurch können Benutzer und Anbieter Komponenten bereitstellen, mit denen die von IBM MQ-Produkten bereitgestellten Komponenten erweitert oder ersetzt werden.

Die mit IBM MQ bereitgestellte Berechtigungsservicekomponente wird als Objektberechtigungsmanager (Object Authority Manager, OAM) bezeichnet. Der OAM wird automatisch für jeden Warteschlangenmanager, den Sie erstellen, aktiviert.

Der OAM verwaltet eine Zugriffssteuerungsliste (Acces Control List, ACL) für jedes IBM MQ-Objekt, dessen Zugriff verwaltet wird. Auf Systemen mit AIX and Linux können nur Gruppen-IDs in einer ACL angezeigt werden. Dies bedeutet, dass alle Mitglieder einer Gruppe die gleichen Berechtigungen haben. Unter IBM i IBM i und auf Windows-Systemen können Benutzer-IDs und Gruppen-IDs in einer ACL angezeigt werden. Dies bedeutet, dass Berechtigungen für einzelne Benutzer und Gruppen erteilt werden können.

Eine Einschränkung von 12 Zeichen gilt sowohl für die Gruppe als auch für die Benutzer-ID. Auf UNIX-Plattformen ist die Länge von Benutzer-IDs generell auf 12 Zeichen begrenzt. Unter AIX und Linux wurde dieser Grenzwert erhöht, aber IBM MQ hält sich weiterhin auf allen UNIX-Plattformen an die Beschränkung auf 12 Zeichen. Wenn Sie eine Benutzer-ID mit mehr als 12 Zeichen verwenden, ersetzt IBM MQ diesen Wert durch den Wert "UNKNOWN". Definieren Sie keine Benutzer-ID mit dem Wert " UNKNOWN ".

Der OAM kann einen Benutzer authentifizieren und die entsprechenden Identitätskontextfelder ändern. Sie aktivieren dies, indem Sie in einem MQCONNX-Aufruf eine Verbindungssicherheitsparameterstruktur (MQCSP) angeben. Die Struktur wird an die OAM Authenticate User-Funktion (MQZ_AUTHENTICATE_USER) übergeben, die die entsprechenden Identitätskontextfelder festlegt. Bei einer MQCONNX-Verbindung von einem IBM MQ-Client werden die Informationen in der MQCSP-Struktur an den Warteschlangenmanager übergeben, mit dem der Client über den Clientverbindungs- und Serververbindungskanal eine Verbindung herstellt. Wenn in diesem Kanal Sicherheitsexits definiert sind, wird der MQCSP in jeden Sicherheitsexit übergeben und kann durch den Exit geändert werden. Sicherheitsexits können auch den MQCSP erstellen. Weitere Informationen zur Verwendung von Sicherheitsexits in diesem Kontext finden Sie im Abschnitt Kanalsicherheitsexitprogramme .

Warnung: In einigen Fällen wird das Kennwort in einer MQCSP-Struktur für eine Clientanwendung über ein Netz in Klartext gesendet. Die Informationen im Abschnitt IBM MQCSP-Kennwortschutz erläutern, wie Sie sicherstellen können, dass Clientanwendungskennwörter angemessen geschützt sind.

Auf AIX, Linux, and Windows-Systemen erteilt und widerruft der Steuerbefehl **setmqaut** Berechtigungen und dient zum Verwalten der ACLs. Beispiel:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER +browse +get
```

ermöglicht es den Mitgliedern der Gruppe VOYAGER, Nachrichten in der Warteschlange MOON.EUROPA zu durchsuchen, deren Eigner der Warteschlangenmanager JUPITER ist. Er ermöglicht es den Teildateien, Nachrichten auch aus der Warteschlange abzurufen. Geben Sie den folgenden Befehl ein, um diese Berechtigungen später wieder zu entziehen:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER -browse -get
```

Der Befehl:

```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +put
```

ermöglicht es den Mitgliedern der Gruppe VOYAGER, Nachrichten in jede Warteschlange mit einem Namen einzureihen, der mit den Zeichen MOON. beginnt. MON.* ist der Name eines generischen Profils.Mit einem *generischen Profil* können Sie Berechtigungen für eine Gruppe von Objekten mit einem einzigen **setmqaut** -Befehl erteilen.

Der Steuerbefehl **dspmqaut** ist verfügbar, um die aktuellen Berechtigungen anzuzeigen, die ein Benutzer oder eine Gruppe für ein angegebenes Objekt hat.Der Steuerbefehl **dmpmqaut** ist auch verfügbar, um die aktuellen Berechtigungen anzuzeigen, die generischen Profilen zugeordnet sind.

IBM i Unter IBM i erteilt ein Administrator Berechtigungen mit dem CL-Befehl GRTMQMAUT und entzieht diese mit dem CL-Befehl RVKMQMAUT. Generische Profile können auch verwendet werden.Der CL-Befehl z. B.:

```
GRTMQMAUT MQMNAME(JUPITER) OBJTYPE(*Q) OBJ('MOON.*') USER(VOYAGER) AUT(*PUT)
```

bietet dieselbe Funktion wie das vorherige Beispiel eines **setmqaut** -Befehls. Es ermöglicht den Mitgliedern der Gruppe VOYAGER, Nachrichten in jede Warteschlange zu stellen, deren Name mit den Zeichen `MOON.` beginnt.

IBM i Mit dem CL-Befehl DSPMQMAUT werden die aktuellen Berechtigungen angezeigt, die Benutzer oder Gruppen für ein angegebener Objekt haben. Die CL-Befehle WRKMQMAUT und WRKMQMAUTD stehen auch für die Arbeit mit den aktuellen Berechtigungen, die Objekten und generischen Profilen zugeordnet sind, zur Verfügung.

Wenn Sie keine Berechtigungsprüfungen wünschen, z. B. in einer Testumgebung, können Sie den OAM inaktivieren.

Multi *PCF für den Zugriff auf OAM-Befehle verwenden*

Auf Systemen mit IBM i, AIX, Linux, and Windows können Sie mithilfe von PCF-Befehlen auf OAM-Verwaltungsbefehle zugreifen.

Die PCF-Befehle und die entsprechenden OAM-Befehle lauten wie folgt:

Tabelle 8. PCF-Befehle und die entsprechenden OAM-Befehle

PCF-Befehl	**OAM, Befehl**
Berechtigungsdatensätze anfragen	dmpmqaut
Entitätsberechtigung inquire	dspmqaut
Berechtigungssatz festlegen	setmqaut
Berechtigungssatz löschen	setmqaut mit Option '-remove'

Die Befehle **setmqaut** und **dmpmqaut** sind auf Mitglieder der Gruppe 'mqm' beschränkt. Die funktional entsprechenden PCF-Befehle können von Benutzern in jeder Gruppe ausgeführt werden, denen dsp-und chg-Berechtigungen auf dem Warteschlangenmanager erteilt wurden.

Weitere Informationen zur Verwendung dieser Befehle finden Sie in Introduction to Programmable Command Formats .

z/OS *Berechtigung zum Arbeiten mit IBM MQ-Objekten in z/OS*

Unter z/OS gibt es sieben Kategorien von Berechtigungsprüfungen, die Aufrufen der MQI zugeordnet sind. Sie müssen bestimmte RACF-Profile definieren und diesen Profilen den erforderlichen Zugriff erteilen. Verwenden Sie das Profil *RESLEVEL* , um zu steuern, wie viele Benutzer-IDs überprüft werden.

Die sieben Kategorien der Berechtigschecks, die mit Aufrufen der MQI verknüpft sind:

Verbindungssicherheit
Die Berechtigungsprüfungen, die ausgeführt werden, wenn eine Anwendung eine Verbindung zu einem WS-Manager herstellt.

Warteschlangensicherheit
Die Berechtigungsprüfungen, die ausgeführt werden, wenn eine Anwendung eine Warteschlange öffnet oder eine permanente dynamische Warteschlange löscht.

Prozesssicherheit
Die Berechtigungsprüfungen, die ausgeführt werden, wenn eine Anwendung ein Prozessobjekt öffnet.

Namensliste, Sicherheit
Die Berechtigungsprüfungen, die ausgeführt werden, wenn eine Anwendung ein Namenslistenobjekt öffnet.

Alternative Benutzersicherheit
Die Berechtigungsprüfungen, die ausgeführt werden, wenn eine Anwendung beim Öffnen eines Objekts eine alternative Benutzerberechtigung anfordert.

Kontextsicherheit
Die Berechtigungsprüfungen, die ausgeführt werden, wenn eine Anwendung eine Warteschlange öffnet und angibt, dass sie die Kontextinformationen in den Nachrichten, die sie in die Warteschlange einreiht, festlegen oder übergeben will.

Themensicherheit
Die Berechtigungsprüfungen, die ausgeführt werden, wenn eine Anwendung ein Thema öffnet.

Jede Kategorie der Berechtigungs-Prüfung wird auf die gleiche Weise implementiert, wie die Sicherheit der Befehlssicherheit und die Sicherheit der Befehlsressourcen implementiert sind. Sie müssen bestimmte RACF-Profile definieren und den entsprechenden Benutzer-IDs und Gruppen Zugriff auf diese Profile auf den erforderlichen Ebenen erteilen. Bei der Warteschlangensicherheit bestimmt die Zugriffsebene die Art der Operation, die die Anwendung in einer Warteschlange ausführen kann. Für die Kontextsicherheit bestimmt die Zugriffsebene, ob die Anwendung folgende Schritte ausführen kann:

- Alle Kontextfelder übergeben
- Übergeben Sie alle Kontextfelder, und legen Sie die Felder für den Identitätskontext fest
- Alle Kontextfelder übergeben und festlegen

Jede Kategorie der Berechtigungs-Check kann durch Definieren von Schalterprofilen ein-oder ausgeschaltet werden.

Alle Kategorien, mit Ausnahme der Verbindungssicherheit, werden unter der Bezeichnung *API-Ressourcensicherheit* zusammengefasst.

Wenn eine API-Ressourcensicherheitsprüfung als Ergebnis eines MQI-Aufrufs von einer Anwendung unter Verwendung einer Stapelverbindung ausgeführt wird, wird standardmäßig nur eine Benutzer-ID überprüft. Wenn eine Prüfung als Ergebnis eines MQI-Aufrufs aus einer CICS- oder IMS-Anwendung oder aus dem Kanalinitiator ausgeführt wird, werden zwei Benutzer-IDs überprüft.

Wenn Sie ein *RESLEVEL-Profil* definieren, können Sie jedoch steuern, ob null, eine oder zwei Benutzer-IDs überprüft werden. Die Anzahl der Benutzer-IDs, die überprüft werden, wird durch die Benutzer-ID bestimmt, die dem Verbindungstyp zugeordnet ist, wenn eine Anwendung eine Verbindung zum Warteschlangenmanager herstellt, und die Zugriffsebene, die Benutzer-ID für das RESLEVEL-Profil hat. Die Benutzer-ID, die den einzelnen Verbindungstyp zugeordnet ist, lautet:

- Die Benutzer-ID der Verbindungstask für Stapelverbindungen
- Die Benutzer-ID des CICS-Adressraums für CICS-Verbindungen
- Die Benutzer-ID für den Adressraum der IMS-Region für IMS-Verbindungen
- Die Benutzer-ID des Kanalinitiatoradressraums für Kanalinitiatorverbindungen

Weitere Informationen über die Berechtigung für die Arbeit mit IBM MQ-Objekten unter z/OS finden Sie unter „Berechtigung für die Verwaltung von IBM MQ unter z/OS” auf Seite 96.

Sicherheit für fernes Messaging

Dieser Abschnitt befasst sich mit Aspekten der Sicherheit im fernen Messaging.

Sie müssen den Benutzern die Berechtigung zur Verwendung der IBM MQ-Funktionen bereitstellen. Dies ist nach Aktionen organisiert, die in Bezug auf Objekte und Definitionen ausgeführt werden sollen. Beispiel:

- WS-Manager können von berechtigten Benutzern gestartet und gestoppt werden.
- Anwendungen müssen eine Verbindung zum Warteschlangenmanager herstellen und die Berechtigung zum Verwenden von Warteschlangen haben.

- Nachrichtenkanäle müssen von berechtigten Benutzern erstellt und gesteuert werden.
- Objekte werden in Bibliotheken aufbewahrt, und der Zugriff auf diese Bibliotheken kann eingeschränkt werden.

Der Nachrichtenkanalagent an einer fernen Site muss überprüfen, ob die Nachricht, die übermittelt wird, von einem Benutzer mit der Berechtigung dazu stammt, dies an dieser fernen Site zu tun. Da die MCAs außerdem über Remotezugriff gestartet werden können, kann es erforderlich sein, zu überprüfen, ob die fernen Prozesse, die versuchen, Ihre MCAs zu starten, berechtigt sind, dies zu tun. Es gibt vier Möglichkeiten, wie Sie damit umgehen können:

1. Verwenden Sie das PutAuthority-Attribut Ihrer RCVR-, RQSTR-oder CLUSRCVR-Kanaldefinition, um zu steuern, welcher Benutzer für die Berechtigungsprüfungen verwendet wird, wenn eingehende Nachrichten in die Warteschlangen gestellt werden. Weitere Informationen finden Sie in der Beschreibung des Befehls DEFINE CHANNEL in der MQSC-Befehlsreferenz.
2. Implementieren Sie Kanalauthentifizierungsdatensätze, um unerwünschte Verbindungsversuche zurückzuweisen oder um einen MCAUSER-Wert basierend auf den folgenden Angaben zu setzen: der fernen IP-Adresse, der fernen Benutzer-ID, dem definierten Namen des TLS-Subjekts (DN) oder dem Namen des fernen Warteschlangenmanagers.
3. Implementieren Sie die Sicherheitsprüfung *Benutzerexit* , um sicherzustellen, dass der entsprechende Nachrichtenkanal berechtigt ist. Die Sicherheit der Installation, die den entsprechenden Kanal hostet, stellt sicher, dass alle Benutzer ordnungsgemäß autorisiert sind, so dass Sie keine einzelnen Nachrichten überprüfen müssen.
4. Implementieren Sie die *Benutzerexit* -Nachrichtenverarbeitung, um sicherzustellen, dass die einzelnen Nachrichten überprüft werden, um die Berechtigung zu erhalten.

IBM i *Sicherheit von IBM MQ for IBM i-Objekten*

Dieser Abschnitt befasst sich mit Aspekten der Sicherheit im fernen Messaging.

Sie müssen Benutzern die Berechtigung zur Nutzung der IBM MQ for IBM i-Funktionen bereitstellen. Diese Berechtigung ist nach Aktionen organisiert, die in Bezug auf Objekte und Definitionen ausgeführt werden sollen. Beispiel:

- WS-Manager können von berechtigten Benutzern gestartet und gestoppt werden.
- Anwendungen müssen eine Verbindung zum Warteschlangenmanager herstellen und die Berechtigung für die Verwendung von Warteschlangen haben.
- Nachrichtenkanäle müssen von berechtigten Benutzern erstellt und gesteuert werden.

Der Nachrichtenkanalagent an einem fernen Standort muss überprüfen, ob die Nachricht, die übermittelt wird, von einem Benutzer mit der Berechtigung zum An-und Absenden der Nachricht an dieser fernen Site abgeleitet wurde. Da die MCAs außerdem über Remotezugriff gestartet werden können, kann es erforderlich sein, zu überprüfen, ob die fernen Prozesse, die versuchen, Ihre MCAs zu starten, berechtigt sind, dies zu tun. Es gibt vier Möglichkeiten, wie Sie damit umgehen können:

- Dekret in der Kanaldefinition, dass Nachrichten eine zulässige *Kontext* -Berechtigung enthalten müssen, da sie andernfalls gelöscht werden.
- Implementieren Sie Kanalauthentifizierungsdatensätze, um unerwünschte Verbindungsversuche zurückzuweisen, oder um einen MCAUSER-Wert basierend auf einer der folgenden Angaben zu setzen: der fernen IP-Adresse, der fernen Benutzer-ID, dem angegebenen TLS-DN (TLS Distinguished Name) oder dem Namen des fernen Warteschlangenmanagers.
- Implementieren Sie die Sicherheitsüberprüfung des Benutzerexits, um sicherzustellen, dass der entsprechende Nachrichtenkanal berechtigt ist. Die Sicherheit der Installation, die den entsprechenden Kanal hostet, stellt sicher, dass alle Benutzer ordnungsgemäß autorisiert sind, so dass Sie keine einzelnen Nachrichten überprüfen müssen.
- Implementieren Sie die Nachrichtenverarbeitung für Benutzerexits, um sicherzustellen, dass die einzelnen Nachrichten für die Autorisierung überprüft werden.

Im folgenden finden Sie einige Fakten, wie Sicherheitsfunktionen in IBM MQ for IBM i durchgeführt werden:

- Benutzer werden von IBM i ermittelt und authentifiziert.
- WS-Manager-Services, die von Anwendungen aufgerufen werden, werden mit der Berechtigung des Benutzerprofils des Warteschlangenmanagers ausgeführt, jedoch im Prozess des Benutzers.
- WS-Manager-Services, die von Benutzerbefehlen aufgerufen werden, werden mit der Berechtigung des Benutzerprofils des Warteschlangenmanagers ausgeführt.

Linux AIX *Sicherheit von Objekten unter AIX and Linux*

Verwaltungsbenutzer müssen Mitglieder der Gruppe 'mqm' auf Ihrem System sein (einschließlich Root), wenn mit dieser ID die Verwaltungsbefehle von IBM MQ verwendet werden sollen.

Sie sollten amqcrsta immer als die Benutzer-ID "mqm" ausführen.

Benutzer-IDs unter AIX and Linux

Der Warteschlangenmanager konvertiert alle Benutzer-IDs in Großbuchstaben oder in Groß-/Kleinschreibung in Kleinbuchstaben. Der WS-Manager fügt dann die Benutzer-IDs in den Kontextteil einer Nachricht ein oder prüft deren Berechtigung. Berechtigungen basieren daher nur auf IDs in Kleinbuchstaben.

Windows *Sicherheit von Objekten auf Windows-Systemen*

Verwaltungsbenutzer müssen Mitglied der Gruppe 'mqm' und der Administratorgruppe auf Windows-Systemen sein, damit diese ID die Verwaltungsbefehle von IBM MQ verwenden kann.

Benutzer-IDs auf Windows-Systemen

Wenn auf Windows-Systemen *kein Nachrichtenexit installiert ist*, konvertiert der Warteschlangenmanager alle Benutzer-IDs in Großschreibung oder in gemischter Groß- und Kleinschreibung in Kleinschreibung. Der WS-Manager fügt dann die Benutzer-IDs in den Kontextteil einer Nachricht ein oder prüft deren Berechtigung. Berechtigungen basieren daher nur auf IDs in Kleinbuchstaben.

Benutzer-IDs auf mehreren Systemen

Andere Plattformen als AIX, Linux, and Windows-Systeme verwenden in Nachrichten Großbuchstaben für Benutzer-IDs. Damit AIX, Linux, and Windows-Systeme in Nachrichten Benutzer-IDs in Kleinbuchstaben verwenden können, muss der Nachrichtenkanalagent (MCA) die entsprechenden Konvertierungen von alphabetischen Zeichen ausführen.

Damit AIX, Linux, and Windows-Systeme in Nachrichten Benutzer-IDs in Kleinbuchstaben verwenden können, werden die folgenden Konvertierungen durch den Nachrichtenkanalagenten (MCA) auf diesen Plattformen ausgeführt:

An der sendenden Seite
Die alphabetischen Zeichen in allen Benutzer-IDs werden in Großbuchstaben umgesetzt, wenn kein Nachrichtenexit installiert ist.

Auf der empfangenden Seite
Die alphabetischen Zeichen in allen Benutzer-IDs werden in Kleinbuchstaben konvertiert, wenn kein Nachrichtenexit installiert ist.

Die automatische Konvertierungen werden nicht ausgeführt, wenn Sie einen Nachrichtenexit in AIX, Linux, and Windows aus einem anderen Grund bereitstellen.

Angepasster Berechtigungsservice verwenden

IBM MQ stellt einen installierbaren Berechtigungsservice bereit. Sie können auswählen, dass ein alternativer Service installiert werden soll.

Die mit IBM MQ bereitgestellte Berechtigungsservicekomponente wird als OAM (Object Authority Manager, Objektberechtigungsmanager) bezeichnet. Wenn der OAM die von Ihnen benötigten Berechtigungsfunktionen nicht liefert, können Sie Ihre eigene Berechtigungsservicekomponente schreiben. Die installierbaren Servicefunktionen, die von einer Berechtigungsservicekomponente implementiert werden müssen, werden im Abschnitt Referenzinformationen zu installierbarer Serviceschnittstelle beschrieben.

Zugriffssteuerung für Clients

Die Zugriffssteuerung basiert auf Benutzer-IDs. Es können viele Benutzer-IDs zur Verwaltung vorhanden sein, und Benutzer-IDs können in unterschiedlichen Formaten vorliegen. Sie können die Serververbindungskanaleigenschaft MCAUSER auf einen speziellen Benutzer-ID-Wert setzen, der von Clients verwendet werden kann.

Die Zugriffssteuerung in IBM MQ basiert auf Benutzer-IDs. Die Benutzer-ID des Prozesses, der MQI-Aufrufe verarbeitet, wird normalerweise verwendet. Bei MQ-MQI-Clients macht die Serververbindung MCA MQI-Aufrufe im Namen von MQ-MQI-Clients. Sie können eine alternative Benutzer-ID für die Serververbindung MCA auswählen, die für die Herstellung von MQI-Aufrufen verwendet werden soll. Die alternative Benutzer-ID kann entweder mit der Client-Workstation oder mit allen anderen Benutzern, die den Zugriff von Clients organisieren und steuern, zugeordnet werden. Die Benutzer-ID muss über die erforderlichen Berechtigungen verfügen, die sie auf dem Server für die Ausgabe von MQI-Aufrufen zugeordnet hat. Die Auswahl einer alternativen Benutzer-ID ist vorzuziehen, damit Clients MQI-Aufrufe mit der Berechtigung der Serververbindung MCA aufrufen können.

Tabelle 9. Die Benutzer-ID, die von einem Serververbindungskanal verwendet wird.

Benutzer-ID	Bei Verwendung
Die Benutzer-ID, die durch einen Sicherheitsexit festgelegt wird.	Wird verwendet, sofern sie nicht durch eine **CHLAUTH TYPE(BLOCKUSER)** -Regel blockiert wird. Weitere Informationen finden Sie im folgenden Abschnitt „Benutzer-ID in einem Sicherheitsexit festlegen” auf Seite 110 .
Die Benutzer-ID, die durch eine CHLAUTH-Regel festgelegt wird.	Wird verwendet, es sei denn, er wird durch einen Sicherheitsexit außer Kraft gesetzt. Weitere Informationen finden Sie unter Kanalauthentifizierungsdatensätze .
Die Benutzer-ID, die im Attribut **MCAUSER** in der SVRCONN-Kanaldefinition definiert ist.	Wird verwendet, es sei denn, sie wird durch einen Sicherheitsexit oder eine CHLAUTH-Regel außer Kraft gesetzt.
Die Benutzer-ID, die von der Clientmaschine aus geflossen ist.	Wird verwendet, wenn keine Benutzer-ID auf andere Weise festgelegt ist.
Die Benutzer-ID, die den Serververbindungskanal gestartet hat.	Wird verwendet, wenn keine andere Benutzer-ID angegeben ist und keine Clientbenutzer-ID in den Flown einfließt. Weitere Informationen finden Sie im folgenden Abschnitt „Die Benutzer-ID, unter der das Kanalprogramm ausgeführt wird.” auf Seite 110 .

Da die Serververbindung MCA MQI-Aufrufe für ferne Benutzer aufruft, ist es wichtig, die Sicherheitsauswirkungen der MQI-Aufrufe des Serververbindungs-MCA, die MQI-Aufrufe ausgeben, im Namen von fernen Clients zu berücksichtigen und den Zugriff auf eine potenziell große Anzahl von Benutzern zu verwalten.

- Ein Ansatz ist, dass der MCA der Serververbindung MQI-Aufrufe an seine eigene Berechtigung ausgeben kann. Aber Vorsicht, es ist in der Regel unerwünscht für den Server-Verbindung MCA, mit seinen leistungsfähigen Zugriffsmöglichkeiten, MQI-Aufrufe im Namen von Clientbenutzern auszugeben.
- Ein anderer Ansatz ist die Verwendung der Benutzer-ID, die vom Client aus fließt. Der MCA der Serververbindung kann MQI-Aufrufe mit Hilfe der Zugriffsfunktionen der Clientbenutzer-ID ausgeben. Dieser Ansatz stellt eine Reihe von Fragen dar, die zu berücksichtigen sind:

  1. Es gibt verschiedene Formate für die Benutzer-ID auf verschiedenen Plattformen. Dies verursacht manchmal Probleme, wenn sich das Format der Benutzer-ID auf dem Client von den akzeptierbaren Formaten auf dem Server unterscheidet.

2. Es gibt potenziell viele Clients mit unterschiedlichen und sich ändernden Benutzer-IDs. Die IDs müssen auf dem Server definiert und verwaltet werden.
3. Ist die Benutzer-ID vertrauenswürdig? Alle Benutzer-IDs können von einem Client aus, nicht notwendigerweise mit der ID des angemeldeten Benutzers, ausgeführt werden. Der Client kann beispielsweise eine ID mit der vollständigen `mqm` -Berechtigung übergeben, die absichtlich nur aus Sicherheitsgründen auf dem Server definiert wurde.

- Der bevorzugte Ansatz besteht darin, Clientidentifizierungs-Token auf dem Server zu definieren und so die Funktionalität von mit Client verbundenen Anwendungen zu begrenzen. Dies wird in der Regel dadurch erreicht, dass die Eigenschaft MCAUSER des Serververbindungskanals auf einen speziellen Benutzer-ID-Wert gesetzt wird, der von Clients verwendet werden soll, und wenige IDs für die Verwendung durch Clients mit unterschiedlichen Berechtigungsstufen auf dem Server definiert.

Benutzer-ID in einem Sicherheitsexit festlegen

Für IBM MQ MQI clients handelt es sich dem Prozess, der die MQI-Aufrufe ausgibt, und den Nachrichtenkanalagenten (MCA) für die Serververbindung. Die Benutzer-ID, die vom MCA der Serververbindung verwendet wird, ist entweder in den Feldern `MCAUserIdentifier` oder `LongMCAUserIdentifier` der MQCD enthalten. Der Inhalt dieser Felder wird wie folgt festgelegt:

- Alle Werte, die von Sicherheitsexits festgelegt werden
- Die Benutzer-ID vom Client
- MCAUSER (in der Definition des Serververbindungskanals)

Der Sicherheitsexit kann die Werte überschreiben, die für ihn sichtbar sind, wenn er aufgerufen wird.

- Wenn das Attribut "MCAUSER" des Serververbindungskanals auf "Nicht leer" gesetzt ist, wird der MCAUSER-Wert verwendet.
- Wenn das Attribut für den Serververbindungskanal MCAUSER leer ist, wird die vom Client empfangene Benutzer-ID verwendet.
- Wenn das Attribut für den Server-Verbindungskanal MCAUSER leer ist und keine Benutzer-ID vom Client empfangen wird, wird die Benutzer-ID, die den Serververbindungskanal gestartet hat, verwendet.

Der IBM MQ-Client gibt die zugesicherte Benutzer-ID nicht an den Server weiter, wenn auf der Clientseite ein Sicherheitsexit verwendet wird.

Die Benutzer-ID, unter der das Kanalprogramm ausgeführt wird.

Wenn die Benutzer-ID-Felder von der Benutzer-ID abgeleitet werden, die den Serververbindungskanal gestartet hat, wird der folgende Wert verwendet:

- z/OS Für z/OS die Benutzer-ID, die über die Tabelle mit gestarteten z/OS-Prozeduren der gestarteten Task des Kanalinitiators zugeordnet ist.
- Für TCP/IP (nicht z/OS) die Benutzer-ID aus dem Eintrag `inetd.conf` oder die Benutzer-ID, mit der der Listener gestartet wurde.
- Für SNA (nicht z/OS) die Benutzer-ID aus dem SNA-Servereintrag oder (falls kein Eintrag vorhanden ist) die eingehende Verbindungsanforderung bzw. die Benutzer-ID, mit der der Listener gestartet wurde.
- Bei NetBIOS oder SPX die Benutzer-ID, unter der das Empfangsprogramm gestartet wurde.

Wenn Serververbindungskanaldefinitionen vorhanden sind, für die das Attribut MCAUSER leer ist, können Clients diese Kanaldefinition verwenden, um eine Verbindung zum Warteschlangenmanager mit der Zugriffsberechtigung herzustellen, die durch die vom Client angegebene Benutzer-ID bestimmt wird. Dies kann eine Sicherheitsexposition sein, wenn das System, auf dem der Warteschlangenmanager ausgeführt wird, unbefugte Netzverbindungen zulässt. Der IBM MQ -Standardserververbindungskanal (SYSTEM.DEF.SVRCONN) ist das Attribut MCAUSER auf leer gesetzt. Um unbefugten Zugriff zu verhindern, aktualisieren Sie das Attribut MCAUSER der Standarddefinition mit einer Benutzer-ID, mit der nicht auf IBM MQ MQ-Objekte zugegriffen werden kann.

Fall von Benutzer-IDs

Wenn Sie einen Kanal mit `runmqsc` definieren, wird das Attribut MCAUSER in Großbuchstaben geändert, sofern die Benutzer-ID nicht in einfachen Anführungszeichen enthalten ist.

ALW Für Server unter AIX, Linux, and Windows werden die Inhalte des Felds `MCAUserIdentifier`, das vom Client empfangen wird, in Kleinbuchstaben geändert.

IBM i Für Server unter IBM i werden die Inhalte des Felds `LongMCAUserIdentifier`, das vom Client empfangen wird, in Großbuchstaben geändert.

Linux AIX Für Server auf AIX and Linux-Systemen werden die Inhalte des Felds `LongMCAUserIdentifier`, das vom Client empfangen wird, in Kleinbuchstaben geändert.

Standardmäßig ist die Benutzer-ID, die bei Verwendung einer IBM MQ JMS-Bindungsanwendung übergeben wird, die Benutzer-ID für die JVM, auf der die Anwendung ausgeführt wird.

Es ist auch möglich, eine Benutzer-ID über die Methode `createQueueConnection` zu übergeben.

Vertraulichkeit planen

Planen Sie, wie Ihre Daten vertraulich behandelt werden.

Sie können die Vertraulichkeit auf Anwendungsebene oder auf Linkebene implementieren. Sie können TLS verwenden. In diesem Fall müssen Sie die Verwendung digitaler Zertifikate planen. Sie können Kanalexitprogramme auch verwenden, wenn die Standardfunktionen Ihre Anforderungen nicht erfüllen.

Zugehörige Konzepte

„Sicherheit auf Verbindungsebene und Sicherheit auf Anwendungsebene vergleichen” auf Seite 111
Dieses Thema enthält Informationen zu verschiedenen Aspekten der Sicherheit auf Verbindungsebene und Sicherheit auf Anwendungsebene und vergleicht die beiden Sicherheitsstufen.

„Kanalexitprogramme” auf Seite 117
Kanalexitprogramme sind Programme, die an definierten Stellen in der Verarbeitungsreihenfolge eines MCA aufgerufen werden. Benutzer und Anbieter können ihre eigenen Kanalexitprogramme schreiben. Einige werden mit IBM bereitgestellt.

„Kanäle mit SSL/TLS schützen” auf Seite 124
Die TLS-Unterstützung in IBM MQ verwendet das Authentifizierungsdatenobjekt des Warteschlangenmanagers und verschiedene MQSC-Befehle. Sie müssen auch Ihre Verwendung digitaler Zertifikate in Betracht ziehen.

Sicherheit auf Verbindungsebene und Sicherheit auf Anwendungsebene vergleichen

Dieses Thema enthält Informationen zu verschiedenen Aspekten der Sicherheit auf Verbindungsebene und Sicherheit auf Anwendungsebene und vergleicht die beiden Sicherheitsstufen.

Die Sicherheit auf Verbindungsebene und auf Anwendungsebene wird in Abbildung 10 auf Seite 112 dargestellt.

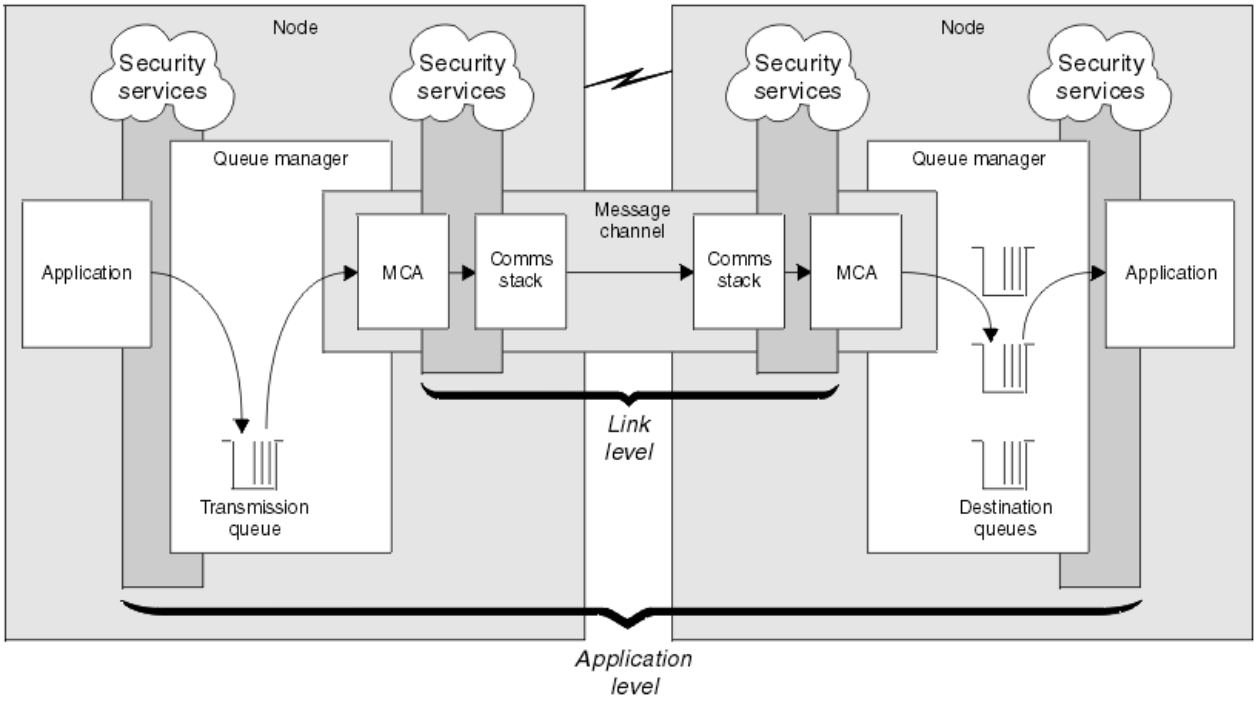


Abbildung 10. Sicherheit auf Verbindungsebene und Sicherheit auf Anwendungsebene

Nachrichten in Warteschlangen schützen

Die Sicherheit auf Verbindungsebene kann Nachrichten schützen, während sie von einem WS-Manager auf einen anderen übertragen werden. Dies ist insbesondere dann wichtig, wenn Nachrichten über ein unsicheres Netz übertragen werden. Sie kann jedoch keine Nachrichten schützen, während sie in Warteschlangen entweder in einem Quellenwarteschlangenmanager, in einem Zielwarteschlangenmanager oder in einem temporären Warteschlangenmanager gespeichert werden.

V 9.2.0 z/OS Die Verschlüsselung des z/OS-Datasets kann einen gewissen Schutz für Nachrichten bereitstellten, die in Warteschlangen gespeichert sind, aber nur für ruhende Daten in einem lokalen Warteschlangenmanager. Weitere Informationen finden Sie im Abschnitt zur Vertraulichkeit für ruhende Daten in IBM MQ for z/OS mit der Dataset-Verschlüsselung. weitere Informationen hierzu.

Die Sicherheit auf Anwendungsebene kann beim Vergleich die Nachrichten schützen, während sie in Warteschlangen gespeichert werden und auch dann angewendet werden, wenn die verteilte Steuerung von Warteschlangen nicht verwendet wird. Dies ist der wesentliche Unterschied zwischen der Sicherheit auf Verbindungsebene und der Sicherheit auf Anwendungsebene und ist in Abbildung 10 auf Seite 112 dargestellt.

Warteschlangenmanager, die nicht in kontrollierten und gesicherten Umgebungen ausgeführt werden

Wenn ein Warteschlangenmanager in einer kontrollierten und gesicherten Umgebung ausgeführt wird, können die von IBM MQ bereitgestellten Verfahren zur Zugriffssteuerung als ausreichend angesehen werden, um die in den Warteschlangen gespeicherten Nachrichten zu schützen. Dies gilt insbesondere dann, wenn es sich nur um eine lokale Warteschlange handelt und die Nachrichten nie den Warteschlangenmanager verlassen. Die Sicherheit auf Anwendungsebene kann in diesem Fall als nicht erforderlich angesehen werden.

Die Sicherheit auf Anwendungsebene kann auch als nicht erforderlich angesehen werden, wenn Nachrichten an einen anderen Warteschlangenmanager übertragen werden, der auch in einer kontrollierten und vertrauenswürdigen Umgebung ausgeführt wird, oder von einem solchen Warteschlangenmanager empfangen werden. Die Sicherheit auf Anwendungsebene wird größer, wenn Nachrichten an einen Warte-

schlangenmanager übertragen oder von einem Warteschlangenmanager empfangen werden, der nicht in einer kontrollierten und vertrauenswürdigen Umgebung ausgeführt wird.

Unterschiedliche Kosten

Die Sicherheit auf Anwendungsebene kann die Sicherheit auf Verbindungsebene in Bezug auf die Verwaltung und die Leistung möglicherweise mehr kosten.

Die Kosten für die Verwaltung sind wahrscheinlich größer, da es potenziell mehr Einschränkungen für die Konfiguration und Verwaltung gibt. Sie müssen z. B. sicherstellen, dass ein bestimmter Benutzer nur bestimmte Nachrichtentypen sendet und Nachrichten nur an bestimmte Ziele sendet. Umgekehrt müssen Sie möglicherweise sicherstellen, dass ein bestimmter Benutzer nur bestimmte Typen von Nachrichten empfängt und Nachrichten nur von bestimmten Quellen empfängt. Anstatt die Sicherheitsservices auf Verbindungsebene in einem einzigen Nachrichtenkanal zu verwalten, müssen Sie möglicherweise Regeln für jedes Paar von Benutzern konfigurieren und verwalten, die Nachrichten über diesen Kanal austauschen.

Es kann Auswirkungen auf die Leistung haben, wenn die Sicherheitsservices jedes Mal aufgerufen werden, wenn eine Anwendung eine Nachricht einreiht oder eine Nachricht abruft.

Organisationen neigen zuerst dazu, die Sicherheit auf Verbindungsebene zu berücksichtigen, da sie möglicherweise einfacher implementiert werden kann. Sie betrachten die Sicherheit auf Anwendungsebene, wenn sie feststellen, dass die Sicherheit auf Verbindungsebene nicht alle ihre Anforderungen erfüllt.

Verfügbarkeit von Komponenten

Im Allgemeinen erfordert ein Sicherheitsservice in einer verteilten Umgebung eine Komponente auf mindestens zwei Systemen. Eine Nachricht kann beispielsweise auf einem System verschlüsselt und auf einem anderen System entschlüsselt werden. Dies gilt sowohl für die Sicherheit auf Verbindungsebene als auch für die Sicherheit auf Anwendungsebene.

In einer heterogenen Umgebung mit verschiedenen Plattformen, die jeweils unterschiedliche Sicherheitsstufen verwenden, sind die erforderlichen Komponenten eines Sicherheitsservice möglicherweise nicht für jede Plattform verfügbar, auf der sie benötigt werden, und in einer Form, die einfach zu verwenden ist. Dies ist wahrscheinlich eher ein Problem für die Sicherheit auf Anwendungsebene als für die Sicherheit auf Verbindungsebene, insbesondere dann, wenn Sie Ihre eigene Sicherheit auf Anwendungsebene durch den Kauf von Komponenten aus verschiedenen Quellen bereitstellen wollen.

Nachrichten in einer Warteschlange für nicht zustellbare Mail

Wenn eine Nachricht durch die Sicherheit auf Anwendungsebene geschützt ist, kann es zu einem Problem kommen, wenn die Nachricht aus irgendeinem Grund nicht an ihr Ziel gelangt und in eine Warteschlange für nicht zustellbare Nachrichten gestellt wird. Wenn Sie nicht herausfinden können, wie die Nachricht aus den Informationen im Nachrichtendeskriptor und dem Header für nicht zustellbare Nachrichten verarbeitet werden kann, müssen Sie möglicherweise den Inhalt der Anwendungsdaten überprüfen. Sie können dies nicht tun, wenn die Anwendungsdaten verschlüsselt sind und nur der vorgesehene Empfänger sie entschlüsseln kann.

Welche Sicherheit auf Anwendungsebene nicht möglich ist

Die Sicherheit auf Anwendungsebene ist keine vollständige Lösung. Selbst wenn Sie die Sicherheit auf Anwendungsebene implementieren, müssen Sie möglicherweise trotzdem einige Sicherheitsservices auf Verbindungsebene benötigen. Beispiel:

- Wenn ein Kanal gestartet wird, kann die gegenseitige Authentifizierung der beiden Nachrichtenkanalagenten dennoch eine Anforderung sein. Dies kann nur durch einen Sicherheitsservice auf Verbindungsebene ausgeführt werden.
- Die Sicherheit auf Anwendungsebene kann den Header der Übertragungswarteschlange (MQXQH), der den eingebetteten Nachrichtendeskriptor enthält, nicht schützen. Sie kann auch in den Datenflüssen

des IBM MQ-Kanalprotokolls nur Nachrichtendaten schützen. Dieser Schutz kann nur durch die Sicherheit auf Verbindungsebene bereitgestellt werden.

- Wenn die Sicherheitsservices auf Anwendungsebene am Serverende eines MQI-Kanals aufgerufen werden, können die Services die Parameter von MQI-Aufrufen, die über den Kanal gesendet werden, nicht schützen. Insbesondere sind die Anwendungsdaten in einem MQPUT-, MQPUT1-oder MQGET-Aufruf nicht geschützt. Nur die Sicherheit auf Verbindungsebene kann den Schutz in diesem Fall gewährleisten.

Sicherheit auf Verbindungsebene

Die *Sicherheit auf Verbindungsebene* bezieht sich auf die Sicherheitsservices, die direkt oder indirekt von einem Nachrichtenkanalsystem, dem Kommunikationssubsystem oder einer Kombination der beiden zusammenarbeitenden Services aufgerufen werden.

Die Sicherheit auf Verbindungsebene ist in Abbildung 10 auf Seite 112 dargestellt.

Im Folgenden finden Sie einige Beispiele für Sicherheitsservices auf Verbindungsebene:

- Der MCA an jedem Ende eines Nachrichtenkanals kann seinen Partner authentifizieren. Dies geschieht, wenn der Kanal gestartet wird und eine DFV-Verbindung hergestellt wurde, aber bevor Nachrichten in den Fluss fließen. Wenn die Authentifizierung an beiden Enden fehlschlägt, wird der Kanal geschlossen, und es werden keine Nachrichten übertragen. Dies ist ein Beispiel für einen Identifizierungs-und Authentifizierungsservice.
- Eine Nachricht kann am sendenden Ende eines Kanals verschlüsselt und an der empfangenden Seite entschlüsselt werden. Dies ist ein Beispiel für einen Vertraulichkeitsdienst.
- Eine Nachricht kann am empfangenden Ende eines Kanals überprüft werden, um festzustellen, ob ihr Inhalt absichtlich geändert wurde, während sie über das Netzwerk übertragen wurde. Dies ist ein Beispiel für einen Datenintegritätsservice.

Von IBM MQ bereitgestellte Sicherheit auf Verbindungsebene

Die wichtigste Funktion zur Bereitstellung der Vertraulichkeit und Datenintegrität in IBM MQ ist die Verwendung von TLS. Weitere Informationen zur Verwendung von TLS in IBM MQ finden Sie unter „TLS-Sicherheitsprotokolle in IBM MQ" auf Seite 26. Für die Authentifizierung stellt IBM MQ die Funktion zur Verwendung von Kanalauthentifizierungsdatensätzen bereit. Kanalauthentifizierungsdatensätze bieten eine präzise Kontrolle über den Zugriff, der für die Verbindung von Systemen erteilt wird, auf der Ebene einzelner Kanäle oder Gruppen von Kanälen. Weitere Informationen finden Sie im Abschnitt „Kanalauthentifizierungssätze" auf Seite 54.

Sicherheit auf eigene Linkebene bereitstellen

Sie können eigene Sicherheitsservices auf Verbindungsebene bereitstellen. Das Schreiben eigener Kanalexitprogramme ist der wichtigste Weg, um eigene Sicherheitsdienste auf Verbindungsebene bereitzustellen.

Kanalexitprogramme werden in „Kanalexitprogramme" auf Seite 117 eingeführt. In diesem Thema wird auch das Kanalexitprogramm beschreiben, das mit IBM MQ for Windows bereitgestellt wird (das SSPI-Kanalexitprogramm). Dieses Kanalexitprogramm wird im Quellenformat bereitgestellt, so dass Sie den Quellcode an Ihre Anforderungen anpassen können. Wenn dieses Kanalexitprogramm oder Kanalexitprogramme, die von anderen Anbietern verfügbar sind, Ihre Anforderungen nicht erfüllen, können Sie Ihre eigenen Anforderungen entwerfen und schreiben. In diesem Thema wird vorgeschlagen, wie Kanalexitprogramme Sicherheitsservices bereitstellen können. Weitere Informationen zum Schreiben eines Kanalexitprogramms finden Sie im Abschnitt Kanalexitprogramme schreiben .

Sicherheit auf Verbindungsebene über einen Sicherheitsexit

Sicherheitsexits arbeiten in der Regel paarweise, d. h. je ein Exit auf jeder Seite eines Kanals. Sie werden unmittelbar nach Abschluss der einleitenden Datenverhandlungen beim Kanalstart aufgerufen.

Sicherheitsexits können zur Identifikation und Authentifizierung, zur Zugriffssteuerung und für den Vertraulichkeitsdienst eingesetzt werden.

Sicherheit auf Verbindungsebene über einen Nachrichtenexit
Ein Nachrichtenexit kann nur für Nachrichtenkanäle, nicht für MQI-Kanäle verwendet werden. Er hat sowohl Zugriff auf den Header der Übertragungswarteschlange (MQXQH), der den eingebetteten Nachrichtendeskriptor enthält, als auch auf die Anwendungsdaten in einer Nachricht. Er kann den Inhalt und die Länge einer Nachricht ändern.

Nachrichtenexits können immer dann eingesetzt werden, wenn ein Zugriff auf die gesamte Nachricht, nicht nur auf Teile davon, erforderlich ist.

Nachrichtenexits können zur Identifikation und Authentifizierung, zur Zugriffssteuerung, für den Vertraulichkeitsdienst, die Datenintegrität sowie den Unbestreitbarkeitsdienst eingesetzt werden, außerdem können sie nicht sicherheitsspezifische Funktionen erfüllen.

Sicherheit auf Verbindungsebene mit Sende-und Empfangsexits
Sende-und Empfangsexits können sowohl für Nachrichten-als auch für MQI-Kanäle verwendet werden. Sie werden für alle Typen von Daten aufgerufen, die auf einem Kanal fließen, und für Flüsse in beide Richtungen.

Sende-und Empfangsexits haben Zugriff auf jedes Übertragungssegment. Sie können ihren Inhalt ändern und seine Länge ändern.

Wenn ein Nachrichtenkanalsystem in einem Nachrichtenkanal eine Nachricht teilen und in mehr als einem Übertragungssegment senden muss, wird für jedes Übertragungssegment, das einen Teil der Nachricht enthält, ein Sendeexit aufgerufen, und am empfangenden Ende wird für jedes Übertragungssegment ein Empfangsexit aufgerufen. Dasselbe gilt für einen MQI-Kanal, wenn die Eingabe-oder Ausgabeparameter eines MQI-Aufrufs zu groß sind, um in einem einzigen Übertragungssegment gesendet zu werden.

In einem MQI-Kanal gibt Byte 10 eines Übertragungssegments den MQI-Aufruf an und gibt an, ob das Übertragungssegment die Eingabe-oder Ausgabeparameter des Aufrufs enthält. Sende-und Empfangsexits können dieses Byte untersuchen, um festzustellen, ob der MQI-Aufruf Anwendungsdaten enthält, die möglicherweise geschützt werden müssen.

Wenn ein Sendeexit zum ersten Mal aufgerufen wird, um alle Ressourcen, die er benötigt, anzufordern und zu initialisieren, kann er den MCA auffordern, einen bestimmten Speicherbereich im Puffer zu reservieren, der ein Übertragungssegment enthält. Wenn es später aufgerufen wird, ein Übertragungssegment zu verarbeiten, kann es diesen Speicherbereich verwenden, um z. B. einen verschlüsselten Schlüssel oder eine digitale Signatur hinzuzufügen. Der entsprechende Empfangsexit am anderen Ende des Kanals kann die durch den Sendeexit hinzugefügten Daten entfernen und ihn zur Verarbeitung des Übertragungssegments verwenden.

Sende-und Empfangsexits eignen sich am besten für Zwecke, in denen sie die Struktur der Daten, die sie verarbeiten, nicht verstehen und daher jedes Übertragungssegment als binäres Objekt behandeln können.

Sende-und Empfangsexits können verwendet werden, um Vertraulichkeit und Datenintegrität zu gewährleisten und andere Verwendungszwecke als die Sicherheit zu verwenden.

Zugehörige Tasks
API-Aufruf in einem Sende-oder Empfangsexitprogramm identifizieren

Sicherheit auf Anwendungsebene
Sicherheit auf Anwendungsebene bezieht sich auf diese Sicherheitsservices, die an der Schnittstelle zwischen einer Anwendung und einem Warteschlangenmanager aufgerufen werden, mit dem sie verbunden ist.

Diese Services werden aufgerufen, wenn die Anwendung MQI-Aufrufe an den WS-Manager ausgibt. Die Services können von der Anwendung, dem Warteschlangenmanager, anderen Produkten, die IBM MQ unterstützen, oder einer Kombination dieser zusammenarbeitenden Komponenten direkt oder indirekt aufgerufen werden. Die Sicherheit auf Anwendungsebene ist in Abbildung 10 auf Seite 112 dargestellt.

Die Sicherheit auf Anwendungsebene wird auch als *End-to-End-Sicherheit* oder *Sicherheit auf Nachrichtenebene* bezeichnet.

Im Folgenden finden Sie einige Beispiele für Sicherheitsservices auf Anwendungsebene:

- Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, enthält der Nachrichtendeskriptor eine Benutzer-ID, die der Anwendung zugeordnet ist. Es sind jedoch keine Daten vorhanden, wie z. B. ein verschlüsseltes Kennwort, das zur Authentifizierung der Benutzer-ID verwendet werden kann. Ein Sicherheitsservice kann diese Daten hinzufügen. Wenn die Nachricht schließlich von der empfangenden Anwendung abgerufen wird, kann eine andere Komponente des Service die Benutzer-ID anhand der Daten authentifizieren, die mit der Nachricht zurückgelegt wurden. Dies ist ein Beispiel für einen Identifizierungs-und Authentifizierungsservice.
- Eine Nachricht kann verschlüsselt werden, wenn sie von einer Anwendung in eine Warteschlange gestellt und entschlüsselt wird, wenn sie von der empfangenden Anwendung abgerufen wird.Dies ist ein Beispiel für einen Vertraulichkeitsdienst.
- Eine Nachricht kann überprüft werden, wenn sie von der empfangenden Anwendung abgerufen wird. Mit dieser Prüfung wird festgelegt, ob der Inhalt absichtlich geändert wurde, da er zum ersten Mal von der sendenden Anwendung in eine Warteschlange gestellt wurde.Dies ist ein Beispiel für einen Datenintegritätsservice.

Advanced Message Security planen
Advanced Message Security (AMS) ist eine Komponente von IBM MQ, die ein hohes Maß an Schutz für sensible Daten bereitstellt, die über das IBM MQ-Netz fließen, während die Endanwendungen nicht imponiert werden.

Wenn Sie hochsensible oder wertvolle Informationen, insbesondere vertrauliche oder zahlungsrelevante Informationen wie Patientenakten oder Kreditkartendaten, verschieben, müssen Sie besonders auf die Informationssicherheit achten. Sicherstellen, dass die Informationen, die sich um das Unternehmen bewegen, seine Integrität erhalten und vor unberechtigtem Zugriff geschützt sind, ist eine ständige Herausforderung und Verantwortung. Es besteht zudem eine hohe Wahrscheinlichkeit, dass Sie zur Einhaltung der Sicherheitsvereinbarungen verpflichtet werden und bei Nichteinhaltung Strafen riskieren.

Sie können Ihre eigenen Sicherheitserweiterungen für IBM MQ entwickeln. Solche Lösungen erfordern jedoch Fachkenntnisse und können kompliziert und kostspielig sein, um sie zu erhalten. Advanced Message Security hilft bei der Bewältigung dieser Aufgaben, die entstehen, wenn Informationen innerhalb des Unternehmens mithilfe nahezu aller Arten von kommerziellen IT-Systemen bewegt werden.

Advanced Message Security erweitert die Sicherheitsfunktionen von IBM MQ folgendermaßen:

- Es stellt End-to-End-Datenschutz auf der Anwendungsebene für Ihre Point-to-Point-Messaging-Infrastruktur mithilfe von Verschlüsselung oder von digitaler Unterzeichnung von Nachrichten zur Verfügung.
- Sie bietet umfassende Sicherheit, ohne den komplexen Sicherheitscode zu schreiben oder vorhandene Anwendungen zu ändern oder neu zu kompilieren.
- Es verwendet die PKI-Technologie (Public Key Infrastructure), um Authentifizierungs-, Berechtigungs-, Vertraulichkeit-und Datenintegritätsservices für Nachrichten bereitzustellen.
- Die Verwaltung von Sicherheitsrichtlinien für Mainframe-Server und verteilte Server wird bereitgestellt.
- Es werden IBM MQ-Server und -Clients unterstützt.
- Es wird in Managed File Transfer integriert, um eine durchgängige und sichere Messaging-Lösung bereitzustellen.

Weitere Informationen finden Sie unter „Advanced Message Security” auf Seite 636.

Bereitstellen der Sicherheit auf Anwendungsebene
Sie können Ihre eigenen Sicherheitsservices auf Anwendungsebene bereitstellen. Damit Sie die Sicherheit auf Anwendungsebene implementieren können, stellt IBM MQ den API-Exit und den API-Steuerübergabeexit bereit.

Der API-Exit und der API-Steuerübergabeexit können die Identifikation und Authentifizierung, die Zugriffssteuerung, die Vertraulichkeit, die Datenintegrität und die Nicht-Repudiationsservices sowie andere Funktionen, die nicht mit der Sicherheit in Zusammenhang stehen, bereitstellen.

Wenn der API-Exit oder der API-Steuerübergabeexit in Ihrer Systemumgebung nicht unterstützt wird, sollten Sie möglicherweise andere Möglichkeiten zur Bereitstellung der Sicherheit auf Anwendungsebene in Betracht ziehen. Eine Möglichkeit besteht darin, eine API einer höheren Ebene zu entwickeln, die die

MQI kapselt. Programmierer verwenden anstelle der MQI dann diese API, um IBM MQ-Anwendungen zu schreiben.

Die häufigsten Gründe für die Verwendung einer API einer höheren Ebene sind:

- So blenden Sie die erweiterten Funktionen der MQI von Programmierern aus.
- Zur Umsetzung von Standards in der Verwendung der MQI.
- So fügen Sie der MQI-Funktion eine Funktion hinzu. Diese zusätzliche Funktion kann Sicherheitsservices sein.

Die Produkte einiger Anbieter verwenden dieses Verfahren, um eine Sicherheit auf Anwendungsebene für IBM MQ bereitzustellen.

Wenn Sie die Sicherheitsservices auf diese Weise bereitstellen möchten, beachten Sie die folgenden Hinweise zur Datenkonvertierung:

- Wenn ein Sicherheitstoken, wie z. B. eine digitale Signatur, zu den Anwendungsdaten in einer Nachricht hinzugefügt wurde, muss jeder Code, der die Datenkonvertierung durchführt, die Anwesenheit dieses Tokens kennen.
- Ein Sicherheitstoken wurde möglicherweise aus einem binären Image der Anwendungsdaten abgeleitet. Daher muss die Überprüfung des Tokens vor dem Konvertieren der Daten erfolgen.
- Wenn die Anwendungsdaten in einer Nachricht verschlüsselt wurden, müssen sie vor der Datenkonvertierung entschlüsselt werden.

Kanalexitprogramme

Kanalexitprogramme sind Programme, die an definierten Stellen in der Verarbeitungsreihenfolge eines MCA aufgerufen werden. Benutzer und Anbieter können ihre eigenen Kanalexitprogramme schreiben. Einige werden mit IBM bereitgestellt.

Es gibt mehrere Typen von Kanalexitprogrammen, aber nur vier haben eine Rolle bei der Bereitstellung der Sicherheit auf Verbindungsebene:

- Sicherheitsexit
- Nachrichtenexit
- Sendeexit
- Empfangsexit

Diese vier Typen von Kanalexitprogrammen sind in Abbildung 11 auf Seite 118 dargestellt und werden in den folgenden Abschnitten beschrieben.

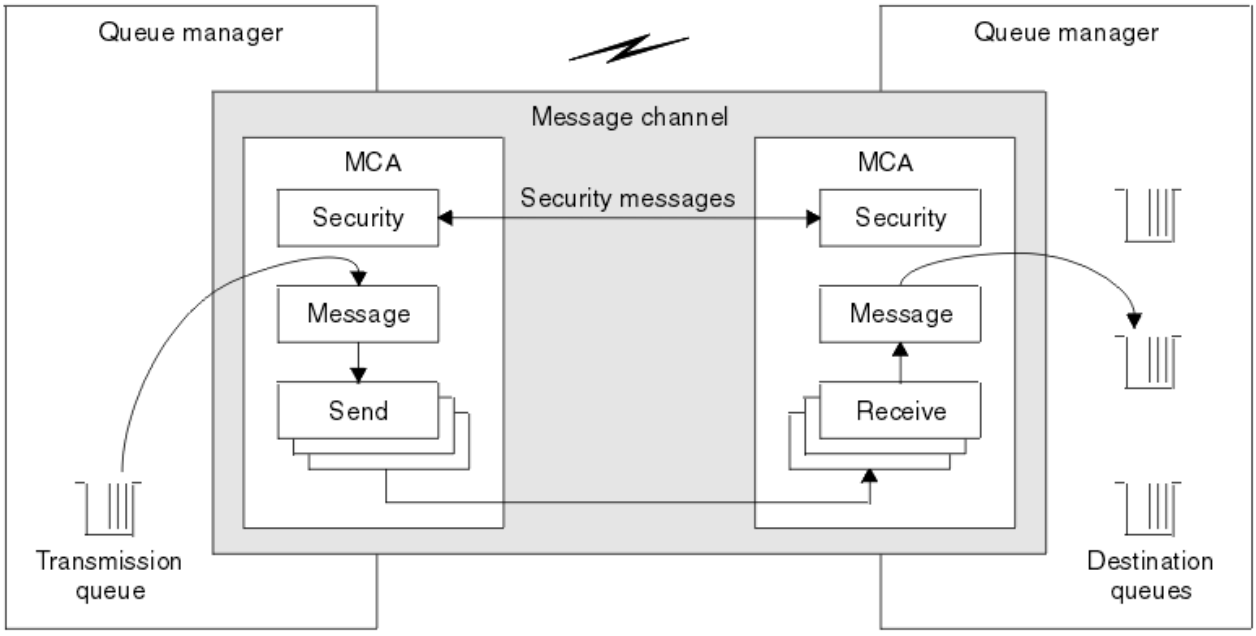


Abbildung 11. Sicherheits-, Nachrichten-, Sende-und Empfangsexits in einem Nachrichtenkanal

Zugehörige Konzepte
Kanalexitprogramme für Messaging-Kanäle

Übersicht über Sicherheitsexits

Sicherheitsexits arbeiten in der Regel paarweise. Sie werden vor der Übertragung von Nachrichten aufgerufen und dienen dem MCA zur Authentifizierung seines Partners.

Sicherheitsexits arbeiten in der Regel paarweise, d. h. je ein Exit auf jeder Seite eines Kanals. Diese Exits werden unmittelbar nach Abschluss der einleitenden Datenverhandlungen beim Kanalstart aufgerufen, jedoch noch vor der Nachrichtenübertragung. Der Sicherheitsexit ist vor allem dazu da, den Nachrichtenkanalagenten auf beiden Kanalseiten die Authentifizierung ihres jeweiligen Partners am anderen Ende zu ermöglichen. Daneben kann ein Sicherheitsexit aber auch noch weitere Funktionen erfüllen, darunter auch solche, die nicht sicherheitsspezifisch sind.

Sicherheitsexits können über *Sicherheitsnachrichten* miteinander kommunizieren. Das Format einer Sicherheitsnachricht ist nicht definiert und wird vom Benutzer festgelegt. Ein mögliches Ergebnis eines Austauschs von Sicherheitsnachrichten ist z. B., dass einer der Sicherheitsexits die Verarbeitung nicht fortsetzt. In diesem Fall wird der Kanal geschlossen, und es werden keine Nachrichten übertragen. Gibt es nur auf einer Seite eines Kanals einen Sicherheitsexit, wird dieser Exit trotzdem aufgerufen; er kann dann entscheiden, ob die Verarbeitung fortgesetzt oder der Kanal geschlossen werden soll.

Sicherheitsexits können für Nachrichten- und MQI-Kanäle aufgerufen werden. Der Name eines Sicherheitsexits wird in Form eines Parameters in der Kanaldefinition auf beiden Seiten eines Kanals angegeben.

Weitere Informationen zu Sicherheitsexits finden Sie unter „Sicherheit auf Verbindungsebene über einen Sicherheitsexit” auf Seite 114.

Nachrichtenexit

Nachrichtenexits werden nur auf Nachrichtenkanälen ausgeführt und funktionieren normalerweise paarweise. Ein Nachrichtenexit kann in der gesamten Nachricht ausgeführt werden und verschiedene Änderungen an ihm vornehmen.

Nachrichtenexits auf der sendenden und empfangenden Seite eines Kanals arbeiten in der Regel paarweise. Ein Nachrichtenexit auf der sendenden Seite eines Kanals wird aufgerufen, nachdem der Nachrichtenkanalnachrichtensender eine Nachricht aus der Übertragungswarteschlange erhalten hat. Am empfan-

genden Ende eines Kanals wird ein Nachrichtenexit aufgerufen, bevor der MCA eine Nachricht in die Zielwarteschlange einreiht.

Ein Nachrichtenexit hat Zugriff auf den Header der Übertragungswarteschlange, MQXQH, der den eingebetteten Nachrichtendeskriptor enthält, und die Anwendungsdaten in einer Nachricht. Ein Nachrichtenexit kann den Inhalt der Nachricht ändern und seine Länge ändern. Eine Änderung der Länge kann das Ergebnis der Komprimierung, Dekomprimierung, Verschlüsselung oder Entschlüsselung der Nachricht sein. Es kann sich auch um das Hinzufügen von Daten zu der Nachricht oder um das Entfernen von Daten aus der Nachricht handeln.

Nachrichtenexits können für jeden Zweck verwendet werden, der Zugriff auf die gesamte Nachricht und nicht einen Teil davon erfordert, und nicht unbedingt für die Sicherheit.

Ein Nachrichtenexit kann feststellen, dass die Nachricht, die gerade verarbeitet wird, nicht weiter an die Zieladresse weiterlaufen soll. Anschließend reiht der Nachrichtenkanalnachrichtennachrichtenkanalnachricht die Nachricht in die Warteschlange für nicht zu Ein Nachrichtenexit kann auch den Kanal schließen.

Nachrichtenexits können nur in Nachrichtenkanälen und nicht in MQI-Kanälen aufgerufen werden. Dies liegt daran, dass ein MQI-Kanal die Eingabe- und Ausgabeparameter von MQI-Aufrufen für Datenflüsse zwischen der IBM MQ MQI client-Anwendung und dem Warteschlangenmanager ermöglichen soll.

Der Name eines Nachrichtenexits wird als Parameter in der Kanaldefinition an jedem Ende eines Kanals angegeben. Sie können auch eine Liste der Nachrichtenexits angeben, die nacheinander ausgeführt werden sollen.

Weitere Informationen zu Nachrichtenexits finden Sie in „Sicherheit auf Verbindungsebene über einen Nachrichtenexit” auf Seite 115.

Sende-und Empfangsexits

Sende-und Empfangsexits funktionieren in der Regel paarweise. Sie arbeiten auf Übertragungssegmenten und werden am besten verwendet, wenn die Struktur der Daten, die sie verarbeiten, nicht relevant ist.

Ein *Sendeexit* an einem Ende eines Kanals und ein *Empfangsexit* am anderen Ende arbeiten normalerweise paarweise. Ein Sendeexit wird unmittelbar vor einem MCA aufgerufen, wenn eine Kommunikation gesendet wird, um Daten über eine DFV-Verbindung zu senden. Ein Empfangsexit wird direkt aufgerufen, nachdem ein MCA die Steuerung nach einem Kommunikationsempfanging wieder aufgenommen hat und Daten von einer DFV-Verbindung empfangen hat. Wenn Dialoge gemeinsam genutzt werden, wird über einen MQI-Kanal eine andere Instanz eines Sende-und Empfangsexits für jede Konversation aufgerufen.

Die Daten, die in Zusammenhang mit dem IBM MQ-Kanalprotokoll zwischen zwei Nachrichtenkanalagenten über einen Nachrichtenkanal ausgetauscht werden, enthalten sowohl Steuerinformationen als auch Nachrichtendaten. In ähnlicher Weise enthalten die Flüsse in einem MQI-Kanal Steuerinformationen sowie die Parameter von MQI-Aufrufen. Sende-und Empfangsexits werden für alle Arten von Daten aufgerufen.

Nachrichtendaten fließen nur in eine Richtung in einem Nachrichtenkanal, aber in einem MQI-Kanal fließen die Eingabeparameter eines MQI-Anrufs in eine Richtung und die Ausgabeparameter fließen in die andere Richtung. Sowohl in Nachrichten-als auch in MQI-Kanälen werden Steuerinformationen in beide Richtungen fließen. Als Ergebnis können Sende-und Empfangsexits an beiden Enden eines Kanals aufgerufen werden.

Die Einheit der Daten, die in einem einzelnen Fluss zwischen zwei Nachrichtenkanalagenten übertragen wird, wird als *Übertragungssegment* bezeichnet. Sende-und Empfangsexits haben Zugriff auf jedes Übertragungssegment. Sie können ihren Inhalt ändern und seine Länge ändern. Ein Sendeexit darf die ersten 8 Byte eines Übertragungssegments jedoch nicht ändern. Diese 8 Byte gehören zum Header des IBM MQ-Kanalprotokolls. Es gibt auch Einschränkungen, wie viel ein Sendeexit die Länge eines Übertragungssegments erhöhen kann. Insbesondere kann ein Sendeexit seine Länge nicht über das Maximum hinaus erhöhen, das zwischen den beiden MCAs beim Kanalstart ausgehandelt wurde.

Wenn eine Nachricht in einem Nachrichtenkanal zu groß ist, um in einem einzigen Übertragungssegment gesendet zu werden, teilt der sendende MCA die Nachricht und sendet sie in mehr als ein Übertragungssegment. Dies hat zur Folge, dass für jedes Übertragungssegment, das einen Teil der Nachricht enthält,

ein Sendeexit aufgerufen wird, und am empfangenden Ende ein Empfangsexit für jedes Übertragungssegment aufgerufen wird. Der empfangende MCA stellt die Nachricht aus den Übertragungssegmenten wieder her, nachdem sie vom Empfangsexit verarbeitet worden sind.

In ähnlicher Weise werden in einem MQI-Kanal die Ein-oder Ausgabeparameter eines MQI-Aufrufs in mehr als einem Übertragungssegment gesendet, wenn sie zu groß sind. Dies kann z. B. bei einem MQPUT-, MQPUT1-oder MQGET-Aufruf auftreten, wenn die Anwendungsdaten ausreichend groß sind.

Unter Berücksichtigung dieser Überlegungen ist es besser, Sende-und Empfangsexits für Zwecke zu verwenden, in denen sie die Struktur der Daten, die sie verarbeiten, nicht verstehen müssen und daher jedes Übertragungssegment als ein binäres Objekt behandeln können.

Ein Sende-oder Empfangsexit kann einen Kanal schließen.

Die Namen eines Sende-Exits und eines Empfangsexits werden als Parameter in der Kanaldefinition an jedem Ende eines Kanals angegeben. Sie können auch eine Liste der Sendeexits angeben, die nacheinander ausgeführt werden sollen. In ähnlicher Weise können Sie eine Liste der Empfangsexits angeben.

Weitere Informationen zu Sende-und Empfangsexits finden Sie in „Sicherheit auf Verbindungsebene mit Sende-und Empfangsexits” auf Seite 115.

Datenintegrität planen

Planen Sie, wie die Integrität Ihrer Daten beibehalten wird.

Sie können die Datenintegrität auf Anwendungsebene oder auf Linkebene implementieren.

Auf der Anwendungsebene können Sie API-Exitprogramme verwenden, wenn die Standardfunktionen Ihre Anforderungen nicht erfüllen. Sie können Advanced Message Security (AMS) verwenden, um Nachrichten digital zu signieren, damit diese vor einer unbefugten Änderung geschützt sind.

Auf der Linkebene können Sie TLS verwenden. In diesem Fall müssen Sie die Verwendung digitaler Zertifikate planen. Sie können Kanalexitprogramme auch verwenden, wenn die Standardfunktionen Ihre Anforderungen nicht erfüllen.

Zugehörige Konzepte

„Kanäle mit SSL/TLS schützen” auf Seite 124
Die TLS-Unterstützung in IBM MQ verwendet das Authentifizierungsdatenobjekt des Warteschlangenmanagers und verschiedene MQSC-Befehle. Sie müssen auch Ihre Verwendung digitaler Zertifikate in Betracht ziehen.

„Datenintegrität in IBM MQ” auf Seite 25
Sie können einen Datenintegritätsservice verwenden, um festzustellen, ob eine Nachricht geändert wurde.

„Advanced Message Security planen” auf Seite 116
Advanced Message Security (AMS) ist eine Komponente von IBM MQ, die ein hohes Maß an Schutz für sensible Daten bereitstellt, die über das IBM MQ-Netz fließen, während die Endanwendungen nicht imponiert werden.

Zugehörige Verweise

API-Exitreferenz

Kanalexitaufrufe und Datenstrukturen

Planung der Prüfung

Entscheiden Sie, welche Daten geprüft werden müssen, und wie Sie Prüfinformationen erfassen und verarbeiten. Überlegen Sie, wie Sie überprüfen können, ob Ihr System ordnungsgemäß konfiguriert ist.

Es gibt mehrere Aspekte der Aktivitätsüberwachung. Die Aspekte, die Sie berücksichtigen müssen, werden häufig durch Prüferfordernisse definiert, und diese Anforderungen werden häufig von regulatorischen Standards wie HIPAA (Health Insurance Portability and Accountability Act) oder SOX (Sarbanes-Oxley) gesteuert. IBM MQ stellt Funktionen bereit, die Sie bei der Einhaltung dieser Standards unterstützen sollen.

Überlegen Sie, ob Sie nur an Ausnahmebedingungen interessiert sind oder ob Sie an allen Systemverhalten interessiert sind.

Einige Aspekte der Prüfung können auch als operationelle Überwachung betrachtet werden; eine Unterscheidung für die Prüfung ist, dass Sie häufig historische Daten betrachten und nicht nur Echtzeitwarnungen betrachten. Die Überwachung wird im Abschnitt Überwachung und Leistung behandelt.

Zu prüfbezogene Daten

Berücksichtigen Sie die Typen von Daten oder Aktivitäten, die Sie prüfen müssen, wie in den folgenden Abschnitten beschrieben:

Änderungen an IBM MQ über die IBM MQ-Schnittstellen
Konfigurieren Sie IBM MQ für die Ausgabe von Instrumentierungsereignissen, insbesondere für Befehlsereignisse und Konfigurationsereignisse.

Änderungen an IBM MQ außerhalb der Steuerung
Einige Änderungen können sich auf die Funktionsweise von IBM MQ auswirken, können aber nicht direkt von IBM MQ überwacht werden. Beispiele für solche Änderungen sind Änderungen an den Konfigurationsdateien `mqs.ini`, `qm.ini`und `mqclient.ini`, die Erstellung und Löschung von Queue Managern, die Installation von Binärdateien, wie z. B. Benutzerexitprogramme, und Änderungen an Dateiberechtigungen. Um diese Aktivitäten zu überwachen, müssen Sie Tools verwenden, die auf der Ebene des Betriebssystems ausgeführt werden. Für verschiedene Betriebssysteme sind verschiedene Tools verfügbar und geeignet. Es können auch Protokolle erstellt werden, die von zugeordneten Tools wie *sudo* erstellt wurden.

Betriebssteuerung von IBM MQ
Möglicherweise müssen Sie Betriebssystemtools verwenden, um Aktivitäten wie das Starten und Stoppen von Warteschlangenmanagern zu prüfen. In einigen Fällen kann IBM MQ für die Ausgabe von Instrumentierungsereignissen konfiguriert werden.

Anwendungsaktivität in IBM MQ
Wenn Sie die Aktionen von Anwendungen prüfen möchten, beispielsweise das Öffnen von Warteschlangen und das Einreihen und Abrufen von Nachrichten, konfigurieren Sie IBM MQ für die Ausgabe der entsprechenden Ereignisse.

Intruder-Alerts
Um versuchte Verstöße gegen die Sicherheitsfunktion zu prüfen, konfigurieren Sie Ihr System so, dass Berechtigungsereignisse ausgegeben werden. Kanalereignisse können auch nützlich sein, um Aktivitäten anzuzeigen, insbesondere dann, wenn ein Kanal unerwartet beendet wird.

Planung der Erfassung, Anzeige und Archivierung von Prüfdaten

Viele der von Ihnen benötigten Elemente werden als IBM MQ-Ereignisnachrichten gemeldet. Sie müssen Tools auswählen, die diese Nachrichten lesen und formatieren können. Wenn Sie an einer Langzeitspeicherung und -analyse interessiert sind, müssen Sie sie in einen Zusatzspeichermechanismus (z. B. eine Datenbank) verschieben. Wenn Sie diese Nachrichten nicht verarbeiten, verbleiben sie in der Ereigniswarteschlange und füllen möglicherweise die Warteschlange aus. Sie können sich entscheiden, ein Tool zu implementieren, das basierend auf einigen Ereignissen automatisch Maßnahmen ergreift, z. B. um einen Alert auszugeben, wenn ein Sicherheitsfehler auftritt.

Überprüfen, ob Ihr System ordnungsgemäß konfiguriert ist

Eine Gruppe von Test werden mit dem IBM MQ Explorer bereitgestellt. Verwenden Sie diese Option, um Ihre Objektdefinitionen auf Probleme zu überprüfen.

Überprüfen Sie außerdem in regelmäßigen Abständen, ob die Systemkonfiguration wie erwartet ausgeführt wird. Obwohl Befehls-und Konfigurationsereignisse berichten können, wenn etwas geändert wird, ist es auch sinnvoll, einen Speicherauszug der Konfiguration zu erstellen und diese mit einer bekannten guten Kopie zu vergleichen.

Planungssicherheit nach Topologie

Dieser Abschnitt behandelt die Sicherheit in bestimmten Situationen, insbesondere für Kanäle, WS-Manager-Cluster, Publish/Subscribe-Anwendungen und Multicastanwendungen sowie bei Verwendung einer Firewall.

Weitere Informationen finden Sie in den folgenden Unterabschnitten:

Kanalberechtigung

Wenn Sie eine Nachricht über einen Kanal senden oder empfangen, müssen Sie Zugriff auf verschiedene IBM MQ-Ressourcen bereitstellen. Nachrichtenkanalagenten (Message Channel Agents, MCAs) sind im Wesentlichen IBM MQ-Anwendungen, die Nachrichten zwischen Warteschlangenmanagern verschieben und als solche Zugriff auf verschiedene IBM MQ-Ressourcen benötigen, um ordnungsgemäß arbeiten zu können.

Um Nachrichten zur PUT-Zeit für MCAs zu empfangen, können Sie entweder die Benutzer-ID, die dem Nachrichtenkanalagenten zugeordnet ist, oder die Benutzer-ID, die der Nachricht zugeordnet ist, verwenden.

Zur CONNECT-Zeit können Sie die zugesicherte Benutzer-ID einem alternativen Benutzer zuordnen, indem Sie **CHLAUTH** -Kanalauthentifizierungsdatensätze verwenden.

In IBM MQ können Kanäle mit der TLS-Unterstützung geschützt werden.

Die Benutzer-IDs, die sendenden und empfangenden Kanälen zugeordnet sind, mit Ausnahme des Sende-Channels, in dem das MCAUSER-Attribut nicht verwendet wird, benötigen Zugriff auf die folgenden Ressourcen:

- Die Benutzer-ID, die einem sendenden Kanal zugeordnet ist, erfordert Zugriff auf den Warteschlangenmanager, die Übertragungswarteschlange, die Warteschlange für dead-Mail und den Zugriff auf alle anderen Ressourcen, die für Kanalexits erforderlich sind.
- Die MCAUSER-Benutzer-ID eines Empfängerkanals benötigt die Berechtigung *+setall* . Dies liegt daran, dass der Empfängerkanal den vollständigen MQMD-Wert einschließlich aller Kontextfelder mit den Daten, die er vom fernen Senderkanal empfangen hat, erstellen muss. Der WS-Manager setzt daher voraus, dass der Benutzer, der diese Aktivität ausführt, die Berechtigung *+setall* hat. Diese *+setall* -Berechtigung muss dem Benutzer für folgende Berechtigungen erteilt werden:
  - Alle Warteschlangen, in die der Empfängerkanal Nachrichten einreiht.
  - Das WS-Manager-Objekt. Weitere Informationen finden Sie unter Autorisierungen für Kontext .
- Die MCAUSER-Benutzer-ID eines Empfängerkanals, in dem der Ersteller eine COA-Berichtsnachricht angefordert hat, benötigt die Berechtigung *+passid* in der Übertragungswarteschlange, die die Berichtsnachricht zurückgibt. Ohne diese Berechtigung werden AMQ8077-Fehlernachrichten protokolliert.
- Mit der Benutzer-ID, die dem empfangenden Kanal zugeordnet ist, können Sie die Zielwarteschlangen öffnen, um Nachrichten in die Warteschlangen zu stellen. Hierbei handelt es sich um die Message Queuing Interface (MQI), wodurch möglicherweise weitere Zugriffssteuerungsprüfungen vorgenommen werden müssen, wenn der Objektberechtigungsmanager (OAM) von IBM MQ nicht verwendet wird. Sie können angeben, ob die Berechtigungsprüfungen für die Benutzer-ID, die dem MCA zugeordnet ist (wie in diesem Thema beschrieben), oder anhand der Benutzer-ID, die der Nachricht zugeordnet ist (aus dem MQMD-Feld UserIdentifier), durchgeführt werden.

  Für die Kanaltypen, auf die er angewendet wird, gibt der Parameter **PUTAUT** einer Kanaldefinition an, welche Benutzer-ID für diese Prüfungen verwendet wird.
  - Der Kanal verwendet standardmäßig den Service-Account des Warteschlangenmanagers, der über vollständige Verwaltungsrechte verfügt und keine Sonderberechtigungen erfordert.
  - Im Falle von Serververbindungskanälen werden die Verwaltungsverbindungen standardmäßig durch CHLAUTH-Regeln blockiert und erfordern eine explizite Bereitstellung.

- Kanäle des Typs "Receiver", "requester" und "cluster-receiver" ermöglichen die lokale Verwaltung durch einen beliebigen benachbarten Warteschlangenmanager, sofern der Administrator keine Schritte unternimmt, um diesen Zugriff zu beschränken.
- Es ist nicht erforderlich, die Berechtigung *dsp* und *ctrlx* für die MCAUSER-Benutzer-ID eines Empfängerkanals zu erteilen.
- Wenn Sie vor IBM MQ 8.0.0 Fix Pack 4 eine Benutzer-ID verwenden, die nicht über Verwaltungsberechtigungen für IBM MQ verfügt, müssen Sie dieser Benutzer-ID die Berechtigung **dsp** und **ctrlx** für den Kanal erteilen, damit dieser ausgeführt werden kann.

  Ab IBM MQ 8.0.0 Fix Pack 4 werden keine Berechtigungsprüfungen ausgeführt, wenn ein Kanal sich selbst resynchronisiert und Folgenummern korrigiert.

  Wenn Sie jedoch den Befehl RESET CHANNEL manuell absetzen, sind weiterhin **+dsp** und **+ctrlx** in allen Releases erforderlich.

  Achtung: Wenn zur Bestätigung eines Nachrichtenstapels ein Kanal zurückgesetzt werden muss, versucht IBM MQ, den Kanal abzufragen, für den die Berechtigung **+dsp** erforderlich ist.
- Das Attribut MCAUSER wird für den SDR-Kanaltyp nicht verwendet.
- Wenn Sie die Benutzer-ID, die der Nachricht zugeordnet ist, verwenden, ist die Benutzer-ID wahrscheinlich von einem fernen System. Diese ferne Systembenutzer-ID muss vom Zielsystem erkannt werden. Die folgenden Befehle sind Beispiele für den Befehlstyp, den Sie ausgeben können, um eine Berechtigung für eine Benutzer-ID von einem fernen System zu erteilen:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect +inq +setall
```

```
setmqaut -m QMgrName -t chl -n Profile -g GroupName +dsp +ctrlx
```

Dabei ist *Profile* ein Kanal.

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

Dabei steht *Profile* für eine Warteschlange mit einem dead-letter (falls festgelegt).

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

Dabei ist *Profile* eine Liste der berechtigten Warteschlangen.

Achtung: Gehen Sie mit Vorsicht vor, wenn Sie eine Benutzer-ID berechtigen, Nachrichten in die Befehlswarteschlange oder in andere sensible Systemwarteschlangen zu stellen.

Die Benutzer-ID, die dem MCA zugeordnet ist, hängt vom Typ des MCA ab. Es gibt zwei Typen von MCA:

Aufrufender MCA
MCAs, die einen Kanal einleiten. Caller MCAs können als einzelne Prozesse gestartet werden, als Threads des Kanalinitiators oder als Threads eines Prozesspools. Die verwendete Benutzer-ID ist die Benutzer-ID, die dem übergeordneten Prozess (dem Kanalinitiator) zugeordnet ist, oder die Benutzer-ID, die dem Prozess zugeordnet ist, mit dem der MCA gestartet wird.

Responder MCA
Responder-MCAs sind MCAs, die als Ergebnis einer Anforderung von einem aufrufenden MCA gestartet werden. Responder-MCAs können als einzelne Prozesse, als Threads des Listeners oder als Threads in einem Prozesspool gestartet werden. Die Benutzer-ID kann einer der folgenden Typen sein (in dieser Reihenfolge der Vorgabe):

1. Auf APPC kann der aufrufende MCA die Benutzer-ID angeben, die für den Responder-MCA verwendet werden soll. Dies wird als Netzbenutzer-ID bezeichnet und gilt nur für Kanäle, die als einzelne Prozesse gestartet wurden. Legen Sie die Netzbenutzer-ID fest, indem Sie den Parameter USERID der Kanaldefinition verwenden.
2. Wenn der Parameter **USERID** nicht verwendet wird, kann die Kanaldefinition des Responder-MCA die Benutzer-ID angeben, die der MCA verwenden muss. Legen Sie die Benutzer-ID fest, indem Sie den Parameter **MCAUSER** der Kanaldefinition verwenden.

3. Wenn die Benutzer-ID nicht von einer der vorherigen (zwei) Methoden festgelegt wurde, wird die Benutzer-ID des Prozesses verwendet, der den MCA oder die Benutzer-ID des übergeordneten Prozesses (Listener) startet.

Zugehörige Konzepte

„Kanalauthentifizierungssätze” auf Seite 54
Die Zugriffsberechtigungen zum Herstellen von Systemverbindungen auf Kanalebene können mithilfe von Kanalauthentifizierungsdatensätzen gezielter gesteuert werden.

Zugehörige Verweise

Eigenschaften des Kanalauthentifizierungsdatensatzes

Kanalinitiatordefinitionen schützen

Nur Mitglieder der Gruppe mqm können Kanalinitiatoren bearbeiten.

IBM MQ-Kanalinitiatoren sind keine IBM MQ-Objekte; der Zugriff wird nicht vom OAM gesteuert. IBM MQ erlaubt Benutzern oder Anwendungen die Bearbeitung dieser Objekte nur, wenn die zugehörige Benutzer-ID ein Mitglied der Gruppe 'mqm' ist. Wenn Sie eine Anwendung haben, die den PCF-Befehl **`StartChannelInitiator`** absetzt, muss die im Nachrichtendeskriptor der PCF-Nachricht angegebene Benutzer-ID Mitglied der Gruppe 'mqm' auf dem Ziel-WS-Manager sein.

Eine Benutzer-ID muss auch ein Mitglied der Gruppe 'mqm' auf der Zielmaschine sein, um die entsprechenden MQSC-Befehle über den Escape-PCF-Befehl auszugeben oder `runmqsc` im indirekten Modus zu verwenden.

Übertragungswarteschlangen

Ferne Nachrichten werden von den Warteschlangenmanagern automatisch in eine Übertragungswarteschlange eingereiht, es ist keine Sonderberechtigung erforderlich.

Wenn Sie eine Nachricht allerdings direkt in eine Übertragungswarteschlange einreihen wollen, ist eine gesonderte Berechtigung erforderlich (siehe Tabelle 12 auf Seite 143).

Kanalexits

Wenn Kanalauthentifizierungsdatensätze nicht geeignet sind, können Sie Kanalexits für hinzugefügte Sicherheit verwenden. Ein Sicherheitsexit stellt eine sichere Verbindung zwischen zwei Sicherheitsexitprogrammen dar. Ein Programm ist für den sendenden Nachrichtenkanalagenten (MCA) und ein Programm für den empfangenden MCA.

Weitere Informationen zu Kanalexits finden Sie in „Kanalexitprogramme” auf Seite 117 .

Kanäle mit SSL/TLS schützen

Die TLS-Unterstützung in IBM MQ verwendet das Authentifizierungsdatenobjekt des Warteschlangenmanagers und verschiedene MQSC-Befehle. Sie müssen auch Ihre Verwendung digitaler Zertifikate in Betracht ziehen.

Digitale Zertifikate und Schlüsselrepositorys

Es ist sinnvoll, das Kennsatzattribut des Warteschlangenmanagers (**CERTLABL**) auf den Namen des persönlichen Zertifikats zu setzen, das für die meisten Kanäle verwendet werden soll, und es für Ausnahmen außer Kraft setzen, indem die Zertifikatsbezeichnung auf den Kanälen festgelegt wird, für die unterschiedliche Zertifikate erforderlich sind.

Wenn Sie viele Kanäle mit Zertifikaten benötigen, die sich vom Standardzertifikat auf dem WS-Manager unterscheiden, sollten Sie die Kanäle zwischen mehreren Warteschlangenmanagern teilen oder einen MQIPT-Proxy vor dem Warteschlangenmanager verwenden, um ein anderes Zertifikat zu präsentieren.

Sie können für jeden Kanal ein anderes Zertifikat verwenden. Wenn Sie jedoch zu viele Zertifikate in einem Schlüsselrepository speichern, können Sie die Leistung beim Starten von TLS-Kanälen voraussichtlich negativ beeinflussen. Versuchen Sie, die Anzahl der Zertifikate in einem Schlüsselrepository auf weniger als etwa 50 zu halten und 100 als maximale Anzahl von Zertifikaten zu berücksichtigen, da die GSKit-Leistung mit größeren Schlüsselrepositorys stark abnimmt.

Die Wahrscheinlichkeit, dass mehrere Zertifikate auf demselben Warteschlangenmanager zulässig sind, erhöht die Wahrscheinlichkeit, dass mehrere CA-Zertifikate auf demselben Warteschlangenmanager verwendet werden. Dies erhöht die Wahrscheinlichkeit, dass die Zertifikatsunterscheidungs-Namespace-Klassenklassenkollisionen für Zertifikate, die von separaten Zertifizierungsstellen ausgestellt wurden, in Konflikt stehen

Während professionelle Zertifizierungsstellen wahrscheinlich vorsichtiger sind, haben die internen Zertifizierungsstellen oft keine klaren Namenskonventionen und Sie könnten mit unbeabsichtigten Übereinstimmungen zwischen einer CA und einer anderen Seite enden.

Sie sollten zusätzlich zum Namen des Zertifikatstinguished Name den Zertifikatausscheidenamen überprüfen. Verwenden Sie hierzu einen SSLPEERMAP-Datensatz für die Kanalauthentifizierung und setzen Sie die Felder **SSLPEER** und **SSLCERTI** so, dass sie mit dem registrierten Namen des Zertifikatregistrierungs-DN bzw. des registrierten Ausstellers übereinstimmen.

Selbst signierte und CA-signierte Zertifikate

Es ist wichtig, die Verwendung digitaler Zertifikate zu planen, wenn Sie Ihre Anwendung entwickeln und testen, und für die Verwendung in der Produktion. Sie können CA-signierte Zertifikate oder selbst signierte Zertifikate verwenden, abhängig von der Verwendung Ihrer Warteschlangenmanager und Clientanwendungen.

Von der Zertifizierungsstelle signierte Zertifikate
Für Produktionssysteme erhalten Sie Ihre Zertifikate von einer anerkannten Zertifizierungsstelle (CA). Wenn Sie ein Zertifikat von einer externen Zertifizierungsstelle erhalten, bezahlen Sie den Service.

Selbst signierte Zertifikate
Während Sie Ihre Anwendung entwickeln, können Sie selbst signierte Zertifikate oder Zertifikate verwenden, die von einer lokalen Zertifizierungsinstanz ausgestellt werden, abhängig von der Plattform:

ALW Auf AIX, Linux, and Windows-Systemen können Sie selbst signierte Zertifikate verwenden. Anweisungen dazu finden Sie unter „Selbst signiertes persönliches Zertifikat unter AIX, Linux, and Windows erstellen" auf Seite 315.

IBM i Auf IBM i-Systemen können Sie Zertifikate verwenden, die von der lokalen Zertifizierungsstelle signiert sind. Anweisungen dazu finden Sie unter „Serverzertifikat unter IBM i anfordern" auf Seite 299.

z/OS Unter z/OS können Sie selbst signierte oder von einer lokalen Zertifizierungsstelle signierte Zertifikate verwenden. Anweisungen hierzu finden Sie unter „Selbst signiertes persönliches Zertifikat unter z/OS erstellen" auf Seite 345 oder „Persönliches Zertifikat unter z/OS anfordern" auf Seite 346.

Selbst signierte Zertifikate sind aus den folgenden Gründen nicht für die Produktionsverwendung geeignet:

- Selbst signierte Zertifikate können nicht widerrufen werden, was es einem Angreifer ermöglicht, eine Identität zu spoen, nachdem ein privater Schlüssel beeinträchtigt wurde. CAs können ein kompromitveres Zertifikat widerrufen, das seine weitere Verwendung verhindert. CA-signierte Zertifikate sind daher sicherer in einer Produktionsumgebung zu verwenden, obwohl selbst signierte Zertifikate für ein Testsystem komfortabler sind.
- Selbst signierte Zertifikate laufen nie ab. Dies ist sowohl praktisch als auch sicher in einer Testumgebung, aber in einer Produktionsumgebung lässt sie sie offen für eventuelle Sicherheitsverletzungen. Das Risiko wird durch die Tatsache verstärkt, dass selbst signierte Zertifikate nicht widerrufen werden können.
- Ein selbst signiertes Zertifikat wird sowohl als persönliches Zertifikat als auch als Stammzertifikat (oder Trust-Anchor) CA-Zertifikat verwendet. Ein Benutzer mit einem selbst signierten persönlichen Zertifikat kann es möglicherweise verwenden, um andere persönliche Zertifikate zu signieren. Im Allgemeinen gilt dies nicht für persönliche Zertifikate, die von einer Zertifizierungsstelle ausgestellt wurden, und stellt eine signifikante Exposition dar.

CipherSpecs und digitale Zertifikate

Nur eine Untergruppe der unterstützten CipherSpecs kann mit allen unterstützten Typen von digitalen Zertifikaten verwendet werden. Es ist daher notwendig, eine geeignete CipherSpec für Ihre digitalen Zertifikate zu wählen. Wenn die Sicherheitsrichtlinie Ihres Unternehmens erfordert, dass eine bestimmte CipherSpec verwendet werden muss, müssen Sie geeignete digitale Zertifikate erwerben.

Weitere Informationen über die Beziehung zwischen CipherSpecs und digitalen Zertifikaten finden Sie unter „Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ" auf Seite 49

Richtlinien zur Zertifikatsprüfung

Der Standard IETF RFC 5280 gibt eine Reihe von Zertifikatvalidierungsregeln an, die die Anwendungssoftware implementieren muss, um Angriffsattacken zu verhindern. Eine Gruppe von Zertifikationsvalidierungsregeln wird als Validierungsrichtlinie für Zertifikate bezeichnet. Weitere Informationen zu Zertifikatsprüfrichtlinien finden Sie in IBM MQ finden Sie im Abschnitt „Zertifikatsprüfrichtlinien in IBM MQ" auf Seite 48.

Prüfung der Zertifikatswiderrufsprüfung planen

Wenn mehrere Zertifikate von verschiedenen Zertifizierungsstellen zulässig sind, kann es zu einer unnötigen zusätzlichen Überprüfung des Zertifikatwiderrufs führen.

Wenn Sie insbesondere die Verwendung eines Widerrufservers von einer bestimmten Zertifizierungsstelle explizit konfiguriert haben, z. B. unter Verwendung einer AUTHINFO-oder MQAIR-Struktur (Authentication Information Record), schlägt eine Widerrufsprüfung fehl, wenn sie mit einem Zertifikat einer anderen Zertifizierungsstelle dargestellt wird.

Sie sollten eine explizite Konfiguration des Zertifikatswiderrufservers vermeiden. Stattdessen sollten Sie die implizite Überprüfung aktivieren, wenn jedes Zertifikat seine eigene Aufrufserverposition in einer Zertifikatserweiterung enthält, z. B. CRL Distribution Point oder OCSP AuthorityInfoAccess.

Weitere Informationen finden Sie unter OCSPCheckExtensions und CDPCheckExtensions .

Befehle und Attribute für TLS-Unterstützung

Das TLS-Protokoll (TLS-Transport Layer Security) bietet Kanalsicherheit mit Schutz vor Ausspionieren, Manipulation und Nachahmungen. Mit der IBM MQ-Unterstützung für TLS können Sie in der Kanaldefinition angeben, dass ein bestimmter Kanal die TLS-Sicherheit verwendet. Sie können auch Details zu dem Typ der gewünschten Sicherheit angeben, z. B. den Verschlüsselungsalgorithmus, den Sie verwenden möchten.

- Mit den folgenden MQSC-Befehlen wird TLS unterstützt:

  ALTER AUTHINFO
  Ändert die Attribute eines Authentifizierungsinformationsobjekts.

  AUTHINFO DEFINIER
  Erstellt ein Authentifizierungsinformationsobjekt.

  DELETE AUTHINFO
  Löscht ein Authentifizierungsinformationsobjekt.

  DISPLAY AUTHINFO
  Zeigt die Attribute für ein bestimmtes Authentifizierungsinformationsobjekt an.

- Die folgenden WS-Manager-Parameter unterstützen TLS:

  CERTLABL
  Definiert eine persönliche Zertifikatsbezeichnung, die verwendet werden soll.

  SSLCRLNL
  Das Attribut "SSLCRLNL" gibt eine Namensliste mit Authentifizierungsinformationsobjekten an, die verwendet werden, um Zertifikatwiderrufspositionen zur Verfügung zu stellen, um eine erweiterte TLS-Zertifikatsprüfung zu ermöglichen.

SSLCRYP
Auf AIX, Linux, and Windows-Systemen wird das Attribut **SSLCryptoHardware** des Warteschlangenmanagers festgelegt. Dieses Attribut ist der Name der Parameterzeichenfolge, die Sie zum Konfigurieren der Verschlüsselungshardware verwenden können, die Sie auf Ihrem System haben.

SSLEV
Legt fest, ob eine TLS-Ereignisnachricht gemeldet wird, wenn ein Kanal, der TLS verwendet, keine TLS-Verbindung herstellen kann.

SSLFIPS
Gibt an, ob nur FIPS-zertifizierte Algorithmen verwendet werden sollen, wenn die Verschlüsselung in IBM MQ und nicht in verschlüsselter Hardware ausgeführt wird. Wenn Verschlüsselungshardware konfiguriert ist, werden die vom Hardwareprodukt bereitgestellten Verschlüsselungsmodule verwendet, und diese können FIPS-zertifiziert sein, die auf eine bestimmte Stufe zertifiziert sind. Dies hängt von dem verwendeten Hardwareprodukt ab.

SSLKEYR
Ordnet auf Systemen mit AIX, Linux, and Windows ein Schlüsselrepository einem Warteschlangenmanager zu. Die Schlüsseldatenbank wird in einer Schlüsseldatenbank von *GSKit* gehalten. Mit dem IBM-Global-Security-Kit (GSKit) können Sie TLS-Sicherheitsverfahren auf AIX, Linux, and Windows-Systemen verwenden.

SSLRKEYC
Die Anzahl der Byte, die in einem TLS-Dialog gesendet und empfangen werden sollen, bevor der geheime Schlüssel erneut verhandelt wird. Die Anzahl der Byte enthält Steuerinformationen, die vom MCA gesendet wurden.

- Die folgenden Kanalparameter unterstützen TLS:

CERTLABL
Definiert eine persönliche Zertifikatsbezeichnung, die verwendet werden soll.

SSLCAUTH
Definiert, ob IBM MQ ein Zertifikat vom TLS-Client benötigt und dies überprüft.

SSLCIPH
Gibt die Verschlüsselungsstärke und -funktion (CipherSpec) an, z. B. TLS_RSA_WITH_AES_128_CBC_SHA. Die CipherSpec muss an beiden Enden des Kanals übereinstimmen.

SSLPEER
Gibt den definierten Namen (eindeutige Kennung) der zulässigen Partner an.

In diesem Abschnitt werden die **setmqaut**-, **dspmqaut**-, **dmpmqaut**-, **rcrmqobj**-, **rcdmqimg**-und **dspmqfls** -Befehle zur Unterstützung des Authentifizierungsinformationsobjekts beschrieben. Darüber hinaus werden die Befehle **runmqckm** (iKeycmd) und **runmqakm** für die Verwaltung von Zertifikaten unter AIX, Linux, and Windows beschrieben. Siehe hierzu folgenden Abschnitte:

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- Schlüssel und Zertifikate verwalten

Eine Übersicht über die Kanalsicherheit mit TLS finden Sie unter.

- „TLS-Sicherheitsprotokolle in IBM MQ” auf Seite 26

Ausführliche Informationen zu MQSC-Befehlen, die TLS zugeordnet sind, finden Sie in.

- ALTER AUTHINFO
- DEFINE AUTHINFO

- DELETE AUTHINFO
- DISPLAY AUTHINFO

Ausführliche Informationen zu den PCF-Befehlen, die TLS zugeordnet sind, finden Sie in.

- Authentifizierungsdatenobjekt ändern, kopieren und erstellen
- Authentifizierungsdatenobjekt löschen
- Authentifizierungsdatenobjekt abfragen

z/OS *IBM MQ for z/OS-Serververbindungskanal*

Der IBM MQ for z/OS-Kanal SVRCONN ist nur sicher, wenn die Kanalauthentifizierung implementiert oder ein Sicherheitsexit mithilfe von TLS hinzugefügt wurde. SVRCONN-Kanäle verfügen nicht über einen standardmäßig definierten Sicherheitsexit.

Sicherheitsbedenken

SVRCONN-Kanäle sind nicht wie ursprünglich definiert sicher, z. B. SYSTEM.DEF.SVRCONN. Um einen SVRCONN-Kanal zu sichern, müssen Sie die Kanalauthentifizierung mit dem Befehl SET CHLAUTH konfigurieren oder einen Sicherheitsexit installieren und TLS implementieren.

Sie müssen einen öffentlich zugänglichen Beispielsicherheitsexit verwenden, einen Sicherheitsexit selbst schreiben oder einen Sicherheitsexit erwerben.

Es stehen mehrere Beispiele zur Verfügung, die Sie als guter Ausgangspunkt für das Schreiben Ihres eigenen SVRCONN-Kanalsicherheitsexits verwenden können.

In IBM MQ for z/OS handelt es sich bei dem Member CSQ4BCX3 in der Bibliothek 'hlq.SCSQC37S' um ein Beispiel für einen Sicherheitsexit, der in der Programmiersprache C geschrieben ist. Der Mustercode 'CSQ4BCX3' wird auch in der Bibliothek 'hlq.SCSQAUTH' vorkompiliert.

Sie können den CSQ4BCX3-Beispielexit implementieren, indem Sie den kompilierten Einzeleintrag hlq.SCSQAUTH (CSQ4BCX3) in eine Ladebibliothek kopieren, die der CSQXLIB-DD in Ihrer CHIN-Proc zugeordnet ist. Beachten Sie, dass für das Schlüsselwort CHIN die Ladebibliothek als "Program Controlled" (Programmgesteuert) festgelegt werden muss.

Ändern Sie Ihren SVRCONN-Kanal, um CSQ4BCX3 als Sicherheitsexit festzulegen.

V 9.2.0 Wenn ein Client eine Verbindung mit diesem SVRCONN-Kanal herstellt, wird CSQ4BCX3 eine Authentifizierung mit dem Paar **RemoteUserIdentifier** und **RemotePassword** aus MQCD oder unter IBM MQ for z/OS 9.1.4 mit dem Paar **CSPUserIdPtr** und **CSPPasswordPtr** aus MQCSP vornehmen. Ist die Authentifizierung erfolgreich, wird **RemoteUserIdentifier** in **MCAUserIdentifier** kopiert, wobei der Identitätskontext des Threads geändert wird.

Wenn ein Client für Long Term Support und Continuous Delivery vor IBM MQ for z/OS 9.1.4 eine Verbindung über diesen SVRCONN-Kanal herstellt, authentifiziert sich CSQ4BCX3 mit dem Paar **RemoteUserIdentifier** und **RemotePassword** aus MQCD. Ist die Authentifizierung erfolgreich, wird **RemoteUserIdentifier** in **MCAUserIdentifier** kopiert, wobei der Identitätskontext des Threads geändert wird.

Wenn Sie einen IBM MQ Java-Client schreiben, können Sie Pop-ups verwenden, um den Benutzer abzufragen und MQEnvironment.userID und MQEnvironment.password festzulegen. Diese Werte werden übergeben, wenn die Verbindung hergestellt wird.

Nachdem Sie nun einen funktionalen Sicherheitsexit haben, möchten Sie vermeiden, dass die Benutzer-ID und das Kennwort im Klartext über das Netz übertragen werden, wenn die Verbindung hergestellt wird, wie auch der Inhalt aller nachfolgenden IBM MQ-Nachrichten. Sie können TLS verwenden, um diese einleitenden Verbindungsinformationen sowie den Inhalt aller IBM MQ -Nachrichten zu verschlüsseln.

Beispiel

Zum Sichern des IBM MQ Explorer SVRCONN-Kanals SYSTEM.ADMIN.SVRCONN führen Sie die folgenden Schritte aus:

1. Kopieren Sie hlq.SCSQAUTH (CSQ4BCX3) in eine Ladebibliothek, die der DD-Datei CSQXLIB in der Datei CHINIT Proc zugeordnet ist.
2. Stellen Sie sicher, dass die Ladebibliothek programmgesteuert ist.
3. Ändern Sie das SYSTEM ADMIN.SVRCONN, um den Sicherheitsexit CSQ4BCX3 zu verwenden.
4. Klicken Sie in IBM MQ Explorer mit der rechten Maustaste auf den Namen des z/OS-Warteschlangenmanagers, wählen Sie **Verbindungsdetail** > **Eigenschaften** > **Benutzer-ID** aus und geben Sie Ihre z/OS-Benutzer-ID ein.
5. Stellen Sie eine Verbindung zum z/OS-Warteschlangenmanager her, indem Sie ein Kennwort eingeben.

Weitere Informationen

Damit der Exit CSQ4BCX3 in einer programmgesteuerten Umgebung ausgeführt werden kann, muss alles, was in den CHIN-Adressraum geladen wird, aus einer programmgesteuerten Bibliothek geladen werden, z. B. alle Bibliotheken in STEPLIB und alle Bibliotheken, die auf der DD CSQXLIB angegeben sind. Um Ihre Ladebibliothek als programmgesteuerte Bibliothek festzulegen, geben Sie RACF-Befehle aus. Im folgenden Beispiel ist der Name der Ladebibliothek MY.TEST.LOADLIB.

```
RALTER PROGRAM * ADDMEM('MY.TEST.LOADLIB'//NOPADCHK)
SETROPTS WHEN(PROGRAM)REFRESH
```

Geben Sie den folgenden IBM MQ -Befehl aus, um den SVRCONN-Kanal zur Implementierung von CSQ4BCX3zu ändern:

```
ALTER CHANNEL(SYSTEM ADMIN.SVRCONN) CHLTYPE(SVRCONN) SCYEXIT(CSQ4BCX3)
```

Im obigen Beispiel wird der verwendete SVRCONN-Kanalname SYSTEM ADMIN.SVRCONN verwendet.

Weitere Informationen zu Kanalexits finden Sie unter „Kanalexitprogramme” auf Seite 117.

Zugehörige Tasks

Kanalexitprogramme unter z/OS schreiben

Sicherheitsservices für SNA LU 6.2

SNA LU 6.2 bietet die Verschlüsselung auf Sitzungsebene, die Authentifizierung auf Sitzungsebene und die Authentifizierung auf Datenaustauschebene an.

Anmerkung: Diese Themensammlung setzt voraus, dass Sie über ein grundlegendes Verständnis von Systems Network Architecture (SNA) verfügen. Die andere in diesem Abschnitt genannte Dokumentation enthält eine kurze Einführung in die relevanten Konzepte und Terminologie. Wenn Sie eine umfassendere technische Einführung in SNA benötigen, finden Sie weitere Informationen im Handbuch *Systems Network Architecture Technical Overview* , IBM Form GC30-3073.

SNA LU 6.2 stellt drei Sicherheitsservices bereit:

- Kryptografie auf Sitzungsebene
- Authentifizierung auf Sitzungsebene
- Authentifizierung auf Konversationsebene

Für die Verschlüsselung auf Sitzungsebene und die Authentifizierung auf Sitzungsebene verwendet SNA den Algorithmus *Data Encryption Standard (DES)* . Der DES-Algorithmus ist ein Blockchiffrierungsalgorithmus, der einen symmetrischen Schlüssel zum Verschlüsseln und Entschlüsseln von Daten verwendet. Sowohl der Block als auch der Schlüssel haben eine Länge von 8 Byte.

Kryptografie auf Sitzungsebene
Verschlüsselung auf Sitzungsebene verschlüsselt Sitzungsdaten mit dem DES-Algorithmus und entschlüsselt sie. Es kann daher verwendet werden, um einen Vertraulichkeitsservice auf Verbindungsebene für SNA LU 6.2-Kanäle bereitzustellen.

Logische Einheiten (LUs) können obligatorische (oder erforderliche) Datenverschlüsselungsdaten, selektive Datenverschlüsselung oder keine Datenkryptografie bereitstellen.

In einer *obligatorischen Chiffriersitzung* verschlüsselt eine LU alle abgehenden Datenanforderungseinheiten und entschlüsselt alle ankommenden Datenanforderungseinheiten.

In einer *selektiven Verschlüsselungssitzung* verschlüsselt eine LU nur die Datenanforderungseinheiten, die durch das sendende Transaktionsprogramm (TP) angegeben sind. Die sendende LU signalisiert, dass die Daten verschlüsselt werden, indem ein Indikator in den Anforderungsheader gesetzt wird. Durch die Überprüfung dieses Indikators kann die empfangende LU mitteilen, welche Anforderungseinheiten entschlüsselt werden sollen, bevor sie an den empfangenden TP übergeben werden.

In einen SNA-Netz handelt es sich bei IBM MQ-Nachrichtenkanalagenten (MCA) um Transaktionsprogramme. MCAs fordern keine Verschlüsselung für alle Daten an, die sie senden. Selektive Datenverschlüsselung ist daher keine Option; es ist nur eine obligatorische Datenverschlüsselung oder keine Datenkryptographie in einer Sitzung möglich.

Informationen zum Implementieren der obligatorischen Datenverschlüsselungsdaten finden Sie in der Dokumentation zu Ihrem SNA-Subsystem. In der gleichen Dokumentation finden Sie Informationen zu stärkeren Formen der Verschlüsselung, die möglicherweise auf Ihrer Plattform verwendet werden können, wie beispielsweise die Triple DES-Verschlüsselung mit 24 Byte unter z/OS.

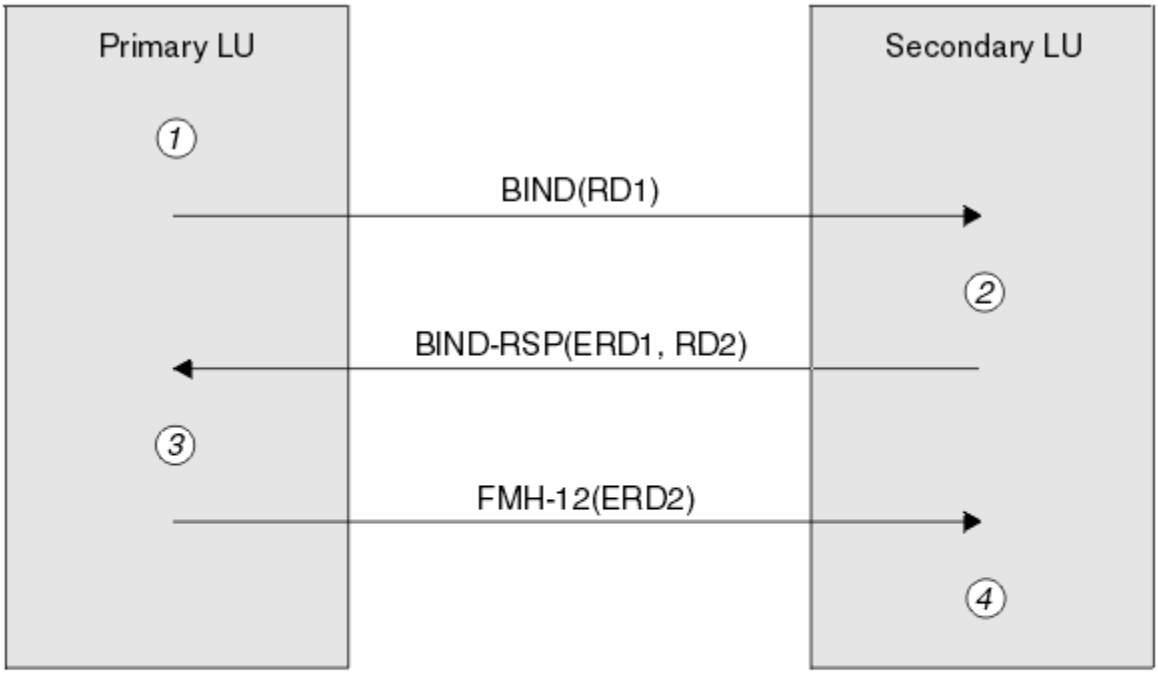
Weitere allgemeine Informationen zur Verschlüsselung auf Sitzungsebene finden Sie im Handbuch *Systems Network Architecture LU 6.2 Reference: Peer Protocols* , IBM Form SC31-6808.

Authentifizierung auf Sitzungsebene
Die *Authentifizierung auf Sitzungsebene* ist ein Sicherheitsprotokoll auf Sitzungsebene, das es zwei LUs ermöglicht, sich gegenseitig zu authentifizieren, während sie eine Sitzung aktivieren. Es wird auch als *LU-LU-Prüfung* bezeichnet.

Da eine LU effektiv das " Gateway " in einem System aus dem Netz ist, können Sie diese Authentifizierungsebene unter bestimmten Umständen als ausreichend ansehen. Wenn Ihr Warteschlangenmanager beispielsweise Nachrichten mit einem fernen Warteschlangenmanager austauschen muss, der in einer kontrollierten und gesicherten Umgebung ausgeführt wird, können Sie möglicherweise darauf vertrauen, dass die Identitäten der verbleibenden Komponenten des fernen Systems nach der Authentifizierung der LU die Identität der verbleibenden Komponenten des fernen Systems vertrauen.

Die Authentifizierung auf Sitzungsebene wird von jeder LU, die das Kennwort des Partners überprüft, erreicht. Das Kennwort wird als *LU-LU-Kennwort* bezeichnet, da zwischen jedem Paar LUs ein Kennwort festgelegt wird. Die Art und Weise, in der ein LU-LU-Kennwort festgelegt wird, ist von der Implementierung abhängig und außerhalb des Geltungsbereichs von SNA.

In Abbildung 12 auf Seite 131 werden die Abläufe für die Authentifizierung auf Sitzungsebene dargestellt.



Legend:

BIND	=	BIND request unit
BIND-RSP	=	BIND response unit
ERD	=	Encrypted random data
FMH-12	=	Function Management Header 12
RD	=	Random data

Abbildung 12. Flows für die Authentifizierung auf Sitzungsebene

Das Protokoll für die Authentifizierung auf Sitzungsebene lautet wie folgt. Die Zahlen in der Prozedur entsprechen den Zahlen in Abbildung 12 auf Seite 131.

1. Die primäre LU generiert einen wahlfreien Datenwert (RD1) und sendet sie in der BIND-Anforderung an die sekundäre LU.
2. Wenn die sekundäre LU die Anforderung BIND mit den Zufallsdaten empfängt, verschlüsselt sie die Daten mit Hilfe des DES-Algorithmus mit ihrer Kopie des LU-LU-Kennworts als Schlüssel. Anschließend generiert die sekundäre LU ebenfalls einen Zufallsdatenwert (RD2), den sie in einer BIND-Antwort zusammen mit den verschlüsselten Daten (ERD1) an die primäre LU sendet.
3. Wenn die primäre LU die BIND-Antwort empfängt, berechnet sie ihre eigene Version der verschlüsselten Daten aus den zufälligen Daten, die sie ursprünglich generiert hat. Dies führt dazu, dass der DES-Algorithmus mit seiner Kopie des LU-LU-Kennworts als Schlüssel verwendet wird. Anschließend vergleicht sie ihre Version mit den verschlüsselten Daten, die sie in der BIND-Antwort empfangen hat. Wenn die beiden Werte identisch sind, weiß die primäre LU, dass die sekundäre LU das gleiche Kennwort hat wie die sekundäre LU und die sekundäre LU authentifiziert wird. Wenn die beiden Werte nicht übereinstimmen, beendet die primäre LU die Sitzung.

   Die primäre LU verschlüsselt dann die zufälligen Daten, die sie in der BIND-Antwort empfangen hat, und sendet die verschlüsselten Daten (ERD2) an die sekundäre LU in einem Funktionsverwaltungs-Header 12 (FMH-12).
4. Wenn die sekundäre LU den FMH-12 empfängt, berechnet sie ihre eigene Version der verschlüsselten Daten aus den zufälligen Daten, die sie generiert hat. Anschließend vergleicht sie ihre Version mit den verschlüsselten Daten, die sie im FMH-12 empfangen hat. Wenn die beiden Werte identisch sind, wird die primäre LU authentifiziert. Wenn die beiden Werte nicht übereinstimmen, beendet die sekundäre LU die Sitzung.

In einer erweiterten Version des Protokolls, die einen besseren Schutz vor dem Menschen in den mittleren Attacken bietet, berechnet die sekundäre LU einen DES-Nachrichtenauthentifizierungscode (MAC) aus RD1, RD2 und den vollständig qualifizierten Namen der sekundären LU, wobei die Kopie des LU-LU-Kennworts als Schlüssel verwendet wird. Die sekundäre LU sendet die MAC an die primäre LU in der BIND-Antwort an Stelle von ERD1.

Die primäre LU authentifiziert die sekundäre LU, indem sie ihre eigene Version des MAC berechnet, die sie mit der in der BIND-Antwort empfangenen MAC-Adresse vergleicht. Die primäre LU berechnet dann eine zweite MAC aus RD1 und RD2 und sendet die MAC an die sekundäre LU im FMH-12 anstelle von ERD2.

Die sekundäre LU authentifiziert die primäre LU, indem sie ihre eigene Version der zweiten MAC-Adresse berechnet, die sie mit der im FMH-12 empfangenen MAC-Adresse vergleicht.

Weitere Informationen zum Konfigurieren der Authentifizierung auf Sitzungsebene finden Sie in der Dokumentation zu Ihrem SNA-Subsystem. Allgemeinere Informationen zur Verschlüsselung auf Sitzungsebene finden Sie im Handbuch *Systems Network Architecture LU 6.2 Reference: Peer Protocols* (SC31-6808).

Authentifizierung auf Konversationsebene
Wenn ein lokales Transaktionsprogramm versucht, einen Datenaustausch mit einem Partner TP zuzuordnen, sendet die lokale LU eine Verbindungsanforderung an die Partner-LU, in der sie aufgefordert wird, den Partner TP zuzuordnen. Unter bestimmten Umständen kann die Zuordnungsanforderung Sicherheitsinformationen enthalten, die von der Partner-LU zur Authentifizierung des lokalen Transaktionsprogramms verwendet werden können. Dies wird als *Authentifizierung auf Konversationsstufe* oder *Endbenutzer-Prüfung* bezeichnet.

In den folgenden Abschnitten wird beschrieben, wie IBM MQ die Unterstützung für die Authentifizierung auf Datenaustauschebene bereitstellt.

Weitere Informationen zur Authentifizierung auf Datenaustauschebene finden Sie im Handbuch *Systems Network Architecture LU 6.2 Reference: Peer Protocols* , IBM Form SC31-6808. Spezifische Informationen für z/OS finden Sie im Handbuch *z/OS MVS Planning: APPC/MVS Management*, SA22-7599.

Weitere Informationen zu CPI-C finden Sie im Handbuch *Common Programming Interface Communications CPI-C Specification* , IBM Form SC31-6180. Weitere Informationen zu APPC/MVS TP Conversation Callable Services finden Sie im Handbuch *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS*, SA22-7621.

Multi *Unterstützung für die Authentifizierung auf Dialogebene auf Multiplattformen*
In diesem Abschnitt erhalten Sie eine Übersicht über die Funktionsweise von Authentifizierungsaufgaben auf Dialogebene auf Multiplattformen.

Die Unterstützung für die Authentifizierung auf Dialogniveau auf Multiplattformen wird in Abbildung 13 auf Seite 133veranschaulicht. Die Zahlen in dem Diagramm entsprechen den Zahlen in der nachfolgenden Beschreibung.

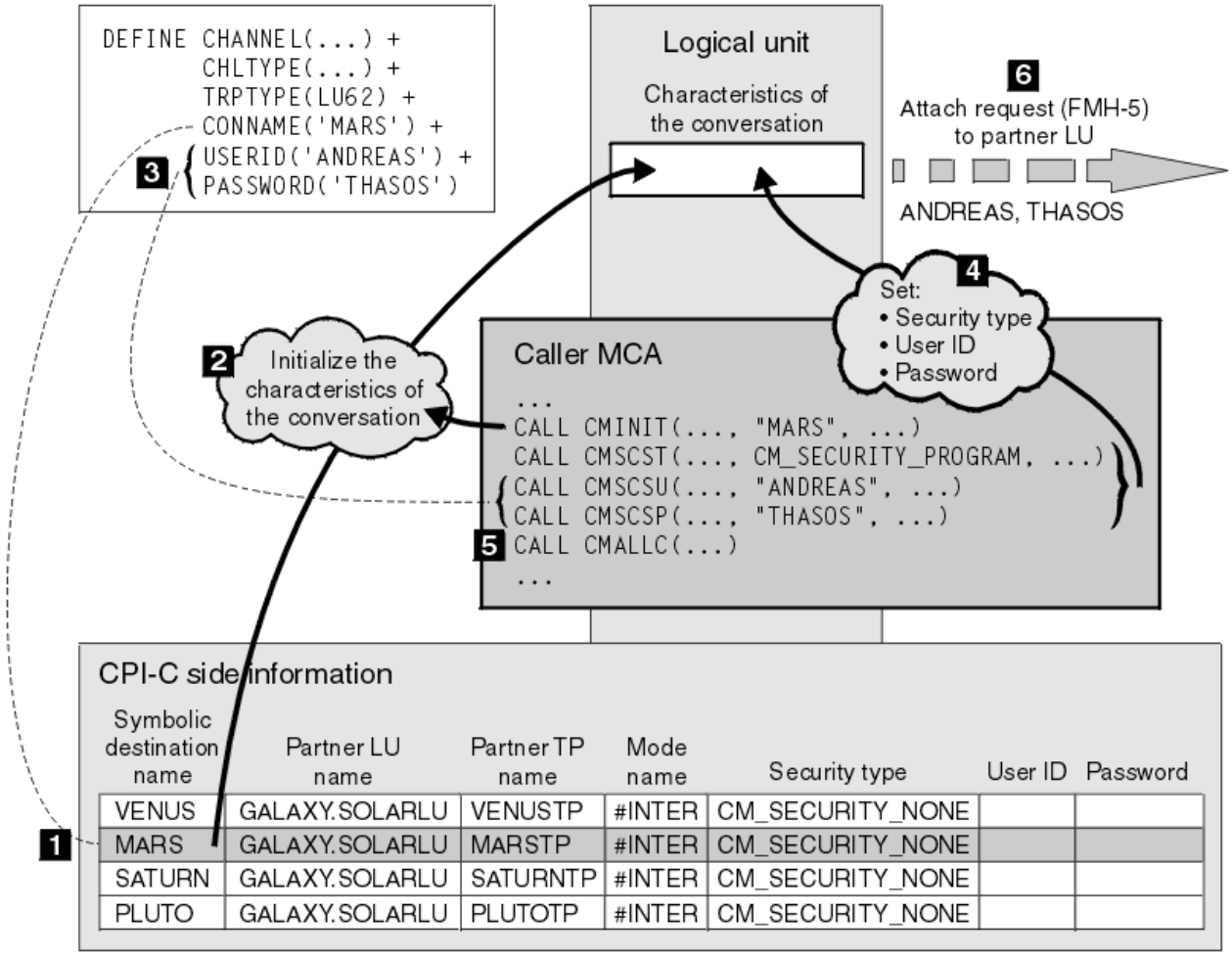


Abbildung 13. IBM MQ-Unterstützung für die Authentifizierung auf Datenaustauschebene

Auf Multiplattformen verwendet ein MCA CPI-C-Aufrufe (Common Programming Interface Communications), um mit einem Partner-MCA über ein SNA-Netz zu kommunizieren. In der Kanaldefinition am Caller-Ende eines Kanals ist der Wert des Parameters CONNAME ein symbolischer Bestimmungsname, der einen CPI-C-Nebeninformationen-Eintrag (1) identifiziert. Dieser Eintrag gibt Folgendes an:

- Der Name der Partner-LU
- Der Name des Partners TP, der ein Responder MCA ist.
- Der Name des Modus, der für den Datenaustausch verwendet werden soll.

Ein Nebeninformationseintrag kann auch die folgenden Sicherheitsinformationen angeben:

- Ein Sicherheitstyp.

  Die allgemein implementierten Sicherheitstypen sind CM_SECURITY_NONE, CM_SECURITY_PROGRAM und CM_SECURITY_SAME, aber andere werden in der CPI-C-Spezifikation definiert.
- Eine Benutzer-ID.
- Ein Kennwort.

Ein aufrufender MCA bereitet einen Dialog mit einem Responder MCA vor, indem er den CPI-C-Aufruf CMINIT absetzt, wobei der Wert von CONNAME als einer der Parameter des Aufrufs verwendet wird. Der CMINIT-Aufruf identifiziert zum Nutzen der lokalen LU den Nebeninformationseintrag, den der MCA für den Datenaustausch zu verwenden beabsichtigt. Die lokale LU verwendet die Werte in diesem Eintrag, um die Merkmale des Datenaustauschs zu initialisieren (2).

Der aufrufende MCA überprüft dann die Werte der Parameter USERID und PASSWORD in der Kanaldefinition (3). Wenn USERID gesetzt ist, gibt der aufrufende MCA die folgenden CPI-C-Aufrufe aus (4):

- CMSCST, um den Sicherheitstyp für den Dialog auf CM_SECURITY_PROGRAM zu setzen.
- CMSCSU, um die Benutzer-ID für den Datenaustausch auf den Wert USERID zu setzen.
- CMSCSP, um das Kennwort für den Datenaustausch auf den Wert von PASSWORD zu setzen. CMSCSP wird nur aufgerufen, wenn PASSWORD festgelegt ist.

Der Sicherheitstyp, die Benutzer-ID und das Kennwort, die durch diese Aufrufe festgelegt werden, überschreiben alle Werte, die zuvor aus dem Nebeninformationen-Eintrag übernommen wurden.

Der aufrufende MCA gibt dann den CPI-C-Aufruf CMALLC aus, um den Dialog zuzuordnen (5). Als Antwort auf diesen Aufruf sendet die lokale LU eine Zuordnungsanforderung (Function Management Header 5, FMH-5) an die Partner-LU (6).

Wenn die Partner-LU eine Benutzer-ID und ein Kennwort akzeptiert, werden die Werte von USERID und PASSWORD in die Zuordnungsanforderung eingeschlossen. Wenn die Partner-LU keine Benutzer-ID und kein Kennwort akzeptiert, sind die Werte nicht in der Anfraganforderung enthalten. Die lokale LU erkennt, ob die Partner-LU eine Benutzer-ID und ein Kennwort als Teil eines Austauschs von Informationen akzeptiert, wenn die LUs eine Sitzung bilden.

In einer späteren Version der Zuordnungsanforderung kann ein Kennwortsubstitut zwischen den LUs anstelle eines eindeutigen Kennworts fließen. Ein Kennwortsubstitut ist ein DES-Nachrichten-Authentifizierungscode (MAC) oder ein SHA-1-Nachrichten-Digest, der aus dem Kennwort gebildet wird. Kennwortsubstitutionen können nur verwendet werden, wenn beide LUs sie unterstützen.

Wenn die Partner-LU eine eingehende Zuordnungsanforderung empfängt, die eine Benutzer-ID und ein Kennwort enthält, kann sie die Benutzer-ID und das Kennwort zum Zweck der Identifikation und Authentifizierung verwenden. Anhand von Zugriffssteuerungslisten kann die Partner-LU auch feststellen, ob die Benutzer-ID über die Berechtigung zum Zuordnen eines Datenaustauschs verfügt und den Responder-MCA zugeordnet hat.

Darüber hinaus kann der Responder-MCA unter der Benutzer-ID ausgeführt werden, die in der Zuordnungsanforderung enthalten ist. In diesem Fall wird die Benutzer-ID zur Standardbenutzer-ID für den Responder-MCA und wird für Berechtigungsprüfungen verwendet, wenn der MCA versucht, eine Verbindung zum WS-Manager herzustellen. Es kann auch dann für Berechtigungsprüfungen verwendet werden, wenn der MCA versucht, auf die Ressourcen des Warteschlangenmanagers zuzugreifen.

Die Art und Weise, in der eine Benutzer-ID und ein Kennwort in einer Zuweisungsanforderung für die Identifikation, Authentifizierung und Zugriffssteuerung verwendet werden können, ist von der Implementierung abhängig. Informationen, die sich speziell auf Ihr SNA-Subsystem beziehen, finden Sie in der entsprechenden Dokumentation.

Wenn USERID nicht festgelegt ist, ruft der aufrufende MCA nicht CMSCST, CMSCSU und CMSCSP auf. In diesem Fall werden die Sicherheitsinformationen, die in einer Zuordnungsanforderung fließen, allein durch die Angaben bestimmt, die im Eintrag für Nebeninformationen angegeben sind und was die Partner-LU akzeptieren wird.

Authentifizierung auf Dialogebene und IBM MQ for z/OS
In diesem Abschnitt finden Sie eine Übersicht über die Funktionsweise der Authentifizierung auf Dialogebene unter z/OS.

Unter IBM MQ for z/OS verwenden MCAs kein CPI-C. Stattdessen verwenden sie APPC/MVS TP Conversation Callable Services, eine Implementierung von Advanced Program-to-Program Communication (APPC), die einige CPI-C-Funktionen enthält. Wenn ein aufrufender MCA einen Datenaustausch zuordnet, wird im Aufruf ein Sicherheitstyp von SAME angegeben. Daher, weil eine APPC/MVS-LU die persistente Überprüfung nur für eingehende Dialoge unterstützt, nicht für abgehende Dialoge, gibt es zwei Möglichkeiten:

- Wenn die Partner-LU die APPC/MVS-LU vertraut und eine bereits überprüfte Benutzer-ID akzeptiert, sendet die APPC/MVS-LU eine Zuordnungsanforderung, die Folgendes enthält:
  - Die Benutzer-ID des Kanalinitiatoradressraums
  - Ein Sicherheitsprofilname, bei dem es sich bei der Verwendung von RACF um den Namen der momentan verbundenen Gruppe mit der Benutzer-ID des Kanalinitiatoradressraums handelt
  - Ein bereits geprüfter Indikator

- Wenn die Partner-LU der APPC/MVS-LU nicht vertraut und eine bereits überprüfte Benutzer-ID nicht akzeptiert, sendet die APPC/MVS-LU eine Zuordnungsanforderung, die keine Sicherheitsinformationen enthält.

Unter IBM MQ for z/OS können die Parameter USERID und PASSWORD im Befehl DEFINE CHANNEL nicht für einen Nachrichtenkanal verwendet werden und sind nur auf der Clientverbindungsseite eines MQI-Kanals gültig. Daher enthält eine Zuordnungsanforderung von einer APPC/MVS-LU nie Werte, die durch diese Parameter angegeben werden.

Sicherheit für WS-Manager-Cluster

Obwohl WS-Manager-Cluster bequem zu verwenden sind, müssen Sie besondere Aufmerksamkeit auf ihre Sicherheit richten.

Ein *WS-Manager-Cluster* ist ein Netz von Warteschlangenmanagern, die logisch in irgendeiner Weise zugeordnet sind. Ein Warteschlangenmanager, der Mitglied eines Clusters ist, wird als *Cluster-WS-Manager* bezeichnet.

Eine Warteschlange, die zu einem Clusterwarteschlangenmanager gehört, kann anderen Warteschlangenmanagern im Cluster bekannt gemacht werden. Eine solche Warteschlange wird als *Clusterwarteschlange* bezeichnet. Jeder WS-Manager in einem Cluster kann Nachrichten an Clusterwarteschlangen senden, ohne dass einer der folgenden Schritte erforderlich ist:

- Eine explizite Definition einer fernen Warteschlange für jede Clusterwarteschlange.
- Explizit definierte Kanäle zu und von jedem fernen WS-Manager
- Eine separate Übertragungswarteschlange für jeden abgehenden Kanal

Sie können einen Cluster erstellen, in dem zwei oder mehr WS-Manager klonen sind. Dies bedeutet, dass sie Instanzen derselben lokalen Warteschlangen haben, einschließlich aller lokalen Warteschlangen, die als Clusterwarteschlangen deklariert sind, und Instanzen derselben Serveranwendungen unterstützen können.

Wenn eine Anwendung, die mit einem Clusterwarteschlangenmanager verbunden ist, eine Nachricht an eine Clusterwarteschlange sendet, die über eine Instanz auf jedem der geklonten Warteschlangenmanager verfügt, legt IBM MQ fest, an welchen Warteschlangenmanager sie gesendet werden soll. Wenn viele Anwendungen Nachrichten an die Clusterwarteschlange senden, verteilt IBM MQ die Auslastung auf alle Warteschlangenmanager, die über eine Instanz der Warteschlange verfügen. Wenn eines der Systeme, auf denen ein geklonter Warteschlangenmanager ausgeführt wird, fehlschlägt, verteilt IBM MQ die Auslastung auf die übrigen Warteschlangenmanager, bis das fehlgeschlagene System erneut gestartet wird.

Wenn Sie WS-Manager-Cluster verwenden, müssen Sie die folgenden Sicherheitsprobleme berücksichtigen:

- Nur ausgewählte WS-Manager zulassen, Nachrichten an Ihren Warteschlangenmanager zu senden
- Nur ausgewählte Benutzer eines fernen Warteschlangenmanagers zulassen, Nachrichten an eine Warteschlange in Ihrem Warteschlangenmanager zu senden
- Anwendungen, die mit Ihrem Warteschlangenmanager verbunden sind, zulassen, Nachrichten nur an ausgewählte ferne Warteschlangen zu senden

Diese Überlegungen sind auch dann relevant, wenn Sie keine Cluster verwenden, aber sie werden wichtiger, wenn Sie Cluster verwenden.

Wenn eine Anwendung Nachrichten an eine Clusterwarteschlange senden kann, kann sie Nachrichten an jede andere Clusterwarteschlange senden, ohne zusätzliche Definitionen für ferne Warteschlangen, Übertragungswarteschlangen oder Kanäle zu benötigen. Es wird daher wichtiger, zu überlegen, ob Sie den Zugriff auf die Clusterwarteschlangen auf Ihrem Warteschlangenmanager einschränken und die Clusterwarteschlangen einschränken müssen, an die Ihre Anwendungen Nachrichten senden können.

Es gibt einige zusätzliche Sicherheitsaspekte, die nur relevant sind, wenn Sie WS-Manager-Cluster verwenden:

- Nur ausgewählten Warteschlangenmanagern die Teilnahme an einem Cluster zulassen
- Unerwünschte WS-Manager zum Verlassen eines Clusters

Weitere Informationen zu allen diesen Aspekten finden Sie im Abschnitt Sichere Cluster schützen .

z/OS Informationen zu speziellen Überlegungen für IBM MQ for z/OS finden Sie unter „Sicherheit in Clustern für Warteschlangenmanager unter z/OS” auf Seite 283.

Zugehörige Tasks

„Verhindern, dass Warteschlangenmanager Nachrichten empfangen” auf Seite 514
Sie können verhindern, dass ein Cluster-WS-Manager Nachrichten empfängt, die er mit Exitprogrammen nicht empfangen kann.

Sicherheit für IBM MQ-Publish/Subscribe

Es gibt zusätzliche Sicherheitsaspekte, die Sie bei der Verwendung von IBM MQ-Publish/Subscribe berücksichtigen müssen.

In einem Publish/Subscribe-System gibt es zwei Arten von Anwendungen: Bereitsteller und Subskribent. *Bereitsteller* liefern Informationen in Form von IBM MQ-Nachrichten. Wenn ein Publisher eine Nachricht veröffentlicht, gibt er ein *Thema* an, das den Betreff der Informationen in der Nachricht identifiziert.

Subskribenten sind die Konsumenten der Informationen, die veröffentlicht werden. Ein Subskribent gibt die Themen an, an denen er interessiert ist, indem er sie subskribiert.

Der *Warteschlangenmanager* ist eine Anwendung, die mit IBM MQ-Publish/Subscribe bereitgestellt wird. Sie empfängt veröffentlichte Nachrichten von Subskribenten und Subskriptionsanforderungen von Subskribenten und leitet die veröffentlichten Nachrichten an die Subskribenten weiter. Ein Subskribent sendet nur Nachrichten zu den Themen, für die er subskribiert hat.

Weitere Informationen finden Sie unter Publish/Subscribe-Sicherheit .

Multicastsicherheit

In diesem Abschnitt finden Sie Informationen dazu, warum Sicherheitsprozesse mit IBM MQ Multicast unter Umständen erforderlich sind.

IBM MQ Multicast verfügt über keine integrierte Sicherheit. Sicherheitsprüfungen werden im Warteschlangenmanager auf MQOPEN-Zeit verarbeitet, und die MQMD-Feldeinstellung wird vom Client verarbeitet. Bei einigen Anwendungen im Netz handelt es sich möglicherweise nicht um IBM MQ-Anwendungen (wie beispielsweise LLM-Anwendungen; weitere Informationen finden Sie unter Multicast-Interoperabilität mit IBM MQ Low Latency Messaging). Deshalb müssen Sie unter Umständen Ihre eigenen Sicherheitsverfahren implementieren, da die empfangenden Anwendungen nicht die Gültigkeit von Kontextfeldern bestätigen können.

Es gibt drei Sicherheitsprozesse, die man in Betracht ziehen kann:

Zugriffssteuerung

Die Zugriffssteuerung in IBM MQ basiert auf Benutzer-IDs. Weitere Informationen zu diesem Thema finden Sie in „Zugriffssteuerung für Clients” auf Seite 109.

Netzsicherheit

Ein isoliertes Netz könnte eine funktionsfähige Sicherheitsoption sein, um gefälschte Nachrichten zu verhindern. Es ist möglich, dass eine Anwendung auf der Multicastgruppenadresse zerstörerische Nachrichten unter Verwendung von nativen Kommunikationsfunktionen veröffentlicht, die nicht von MQ-Nachrichten unterschieden werden können, da sie von einer Anwendung auf derselben Multicastgruppenadresse stammen.

Es ist auch möglich, dass ein Client auf der Multicastgruppenadresse Nachrichten empfängt, die für andere Clients auf derselben Multicastgruppenadresse bestimmt waren.

Durch Isolieren des Multicastnetzes wird sichergestellt, dass nur gültige Clients und Anwendungen Zugriff haben. Diese Sicherheitsvorkehrung kann verhindern, dass heimtückische Nachrichten in die Daten kommen, und vertrauliche Informationen werden nicht mehr angezeigt.

Weitere Informationen zu Netzadressen für Multicastgruppen finden Sie unter Das geeignete Netz für den Multicastverkehr festlegen

Digitale Signaturen

Eine digitale Signatur wird gebildet, indem eine Darstellung einer Nachricht verschlüsselt wird. Die Verschlüsselung verwendet den privaten Schlüssel des Unterzeichners und arbeitet für die Effizienz in der Regel in der Regel in einem Nachrichten-Digest und nicht in der Nachricht selbst. Das digitale Signieren einer Nachricht vor einem MQPUT ist eine gute Sicherheitsvorkehrung, aber dieser Prozess kann sich negativ auf die Leistung auswirken, wenn ein großes Volumen an Nachrichten vorhanden ist.

Digitale Signaturen variieren mit den Daten, die signiert werden. Wenn zwei verschiedene Nachrichten von derselben Entität digital signiert werden, unterscheiden sich die beiden Signaturen voneinander, aber beide Signaturen können mit demselben öffentlichen Schlüssel verifiziert werden, d.

Wie bereits in diesem Abschnitt erwähnt, kann es für eine Anwendung in der Multicastgruppenadresse möglich sein, zerstörerische Nachrichten unter Verwendung von nativen Kommunikationsfunktionen zu veröffentlichen, die nicht von MQ-Nachrichten unterschieden werden können. Digitale Signaturen stellen einen Ursprungsnachweis zur Verfügung, und nur der Absender kennt den privaten Schlüssel, der einen starken Beweis dafür liefert, dass der Absender der Absender der Nachricht ist.

Weitere Informationen zu diesem Thema finden Sie in „Verschlüsselungskonzepte” auf Seite 9.

Firewalls und Internet Pass-Thru

Sie verwenden normalerweise eine Firewall, um den Zugriff von feindlichen IP-Adressen zu verhindern, z. B. in einem Denial of Service-Angriff. Möglicherweise müssen Sie jedoch IP-Adressen in IBM MQ blockieren, während Sie vielleicht darauf warten, dass ein Sicherheitsadministrator die Firewallregeln aktualisiert.

Wenn Sie eine oder mehrere IP-Adressen blockieren möchten, erstellen Sie einen Kanalauthentifizierungsdatensatz des Typs BLOCKADDR oder ADDRESSMAP. Weitere Informationen finden Sie im Abschnitt „Blockieren bestimmter IP-Adressen” auf Seite 410.

Sicherheit für IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru kann die Kommunikation durch eine Firewall vereinfachen, aber dies hat Auswirkungen auf die Sicherheit.

IBM MQ Internet Pass-Thru (MQIPT) ist eine optionale Komponente von IBM MQ, mit der Messaging-Lösungen zwischen fernen Sites über das Internet implementiert werden können.

Mit MQIPT können zwei Warteschlangenmanager Nachrichten austauschen oder eine IBM MQ-Clientanwendung kann über das Internet eine Verbindung zu einem Warteschlangenmanager herstellen, ohne dass eine direkte TCP/IP-Verbindung erforderlich ist. Dies ist nützlich, wenn eine Firewall eine direkte TCP/IP-Verbindung zwischen zwei Systemen verhindert. Die Protokollübertragung durch den IBM MQ-Kanal kann in beiden Richtungen durch eine Firewall vereinfacht und besser verwaltbar werden, indem die Abläufe in HTTP oder durch die Funktion als Proxy gesteuert werden. Mithilfe von Transport Layer Security (TLS) kann es auch zum Verschlüsseln und Entschlüsseln von Nachrichten verwendet werden, die über das Internet gesendet werden.

Wenn Ihr IBM MQ-System mit MQIPT kommuniziert, stellen Sie außer bei der Verwendung des SSL-Proxy-Modus in MQIPT sicher, dass die von IBM MQ verwendete CipherSpec mit der von MQIPT verwendeten Cipher-Suite übereinstimmt:

- Wenn MQIPT als TLS-Server verwendet wird und IBM MQ eine Verbindung als TLS-Client herstellt, muss die von IBM MQ verwendete CipherSpec einer Cipher-Suite entsprechen, die im relevanten MQIPT-Schlüsselring aktiviert ist.

- Wenn MQIPT als TLS-Client ausgeführt wird und eine Verbindung zu einem IBM MQ-TLS-Server herstellt, muss die Cipher-Suite von MQIPT mit der CipherSpec übereinstimmen, die im empfangenden IBM MQ-Kanal definiert ist.

Bei einer Migration von MQIPT auf die integrierte TLS-Unterstützung für IBM MQ übertragen Sie die digitalen Zertifikate aus dem MQIPT-Schlüsselring mithilfe von **mqiptKeyman** oder **mqiptKeycmd**.

Weitere Informationen finden Sie unter IBM MQ Internet Pass-Thru.

z/OS Prüfliste für die Implementierung der IBM MQ for z/OS-Sicherheit

In diesem Abschnitt finden Sie Informationen zum schrittweisen Vorgehen für die Ermittlung und Definition der Sicherheitsimplementierung für jeden Ihrer IBM MQ-Warteschlangenmanager.

RACF stellt Definitionen für die IBM MQ-Sicherheitsklassen in der bereitgestellten statischen Klassendeskriptortabelle (Class Descriptor Table, CDT) bereit. Während Sie die Prüfliste bearbeiten, können Sie feststellen, welche dieser Klassen Ihre Konfiguration erfordert. Sie müssen sicherstellen, dass sie wie im Abschnitt „RACF-Sicherheitsklassen” auf Seite 199 beschrieben aktiviert sind.

Weitere Informationen finden Sie in anderen Abschnitten, insbesondere unter „Profile, die zum Steuern des Zugriffs auf IBM MQ-Ressourcen verwendet werden” auf Seite 210.

Wenn Sie eine Sicherheitsprüfung benötigen, führen Sie die folgende Prüfliste aus, um sie zu implementieren:

1. Aktivieren Sie die RACF-Klasse MQADMIN (Profile in Großschreibung) oder MXADMIN (Profile in in Groß-/Kleinschreibung).
   - Möchten Sie die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange, auf der Ebene des Warteschlangenmanagers oder in einer Kombination aus beiden?

     Siehe „Profile zur Steuerung der Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange oder des Warteschlangenmanagers” auf Seite 204.
2. Benötigen Sie Verbindungssicherheit?
   - **Ja** : Aktivieren Sie die MQCONN-Klasse. Definieren Sie geeignete Verbindungsprofile auf Warteschlangenmanagerebene oder auf Ebene der Gruppe mit gemeinsamer Warteschlange in der MQCONN-Klasse. Erlauben Sie dann den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile.

     Anmerkung: Nur Benutzer der API-Anforderung `MQCONN` oder Benutzer-IDs für den CICS- oder IMS-Adressraum benötigen Zugriff auf das entsprechende Verbindungsprofil.
   - **Nein**: Definieren Sie ein hlq.NO.CONNECT.CHECKS-Profil auf Warteschlangenmanagerebene oder auf der Ebene der Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.
3. Benötigen Sie Sicherheitsprüfungen für Befehle?
   - **Ja** : Aktivieren Sie die MQCMDS-Klasse. Definieren Sie geeignete Befehlsprofile auf Warteschlangenmanagerebene oder auf Ebene der Gruppe mit gemeinsamer Warteschlange in der MQCMDS-Klasse. Erlauben Sie dann den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile.

     Wenn Sie eine Gruppe mit gemeinsamer Warteschlange verwenden, müssen Sie unter Umständen die Benutzer-IDs einschließen, die vom Warteschlangenmanager selbst und vom Kanalinitiator verwendet werden. Weitere Informationen finden Sie unter „IBM MQ for z/OS-Ressourcensicherheit einrichten” auf Seite 273.
   - **Nein**: Definieren Sie ein hlq.NO.CMD.CHECKS-Profil für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.
4. Benötigen Sie Sicherheit für die Ressourcen, die in Befehlen verwendet werden?
   - **Ja** : Stellen Sie sicher, dass die Klasse MQADMIN oder MXADMIN aktiv ist. Definieren Sie geeignete Profile zum Schützen von Ressourcen in Befehlen auf Warteschlangenmanagerebene oder auf Ebene der Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.

Erlauben Sie dann den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile. Setzen Sie den Parameter CMDUSER in CSQ6SYSP auf die Standardbenutzer-ID, die für Befehlssicherheitsüberprüfungen verwendet werden soll.

Wenn Sie eine Gruppe mit gemeinsamer Warteschlange verwenden, müssen Sie unter Umständen die Benutzer-IDs einschließen, die vom Warteschlangenmanager selbst und vom Kanalinitiator verwendet werden. Weitere Informationen finden Sie unter „IBM MQ for z/OS-Ressourcensicherheit einrichten” auf Seite 273.

- **Nein**: Definieren Sie ein hlq.NO.CMD.RESC.CHECKS-Profil für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.

5. Benötigen Sie die Warteschlangensicherheit?
   - **Ja** : Aktivieren Sie die Klasse MQQUEUE oder MXQUEUE. Definieren Sie geeignete Warteschlangenprofile für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQQUEUE oder MXQUEUE. Erlauben Sie dann den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile.
   - **Nein**: Definieren Sie ein hlq.NO.QUEUE.CHECKS-Profil für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.
6. Benötigen Sie Prozesssicherheit?
   - **Ja** : Aktivieren Sie die MQPROC-oder MXPROC-Klasse. Definieren Sie geeignete Prozessprofile auf der Ebene des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange und ermöglichen Sie den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile.
   - **Nein**: Definieren Sie ein hlq.NO.PROCESS.CHECKS-Profil für den entsprechenden Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.
7. Benötigen Sie Namenslistensicherheit?
   - **Ja** : Aktivieren Sie die MQNLIST-oder MXNLISTclass-Klasse. Definieren Sie geeignete Namenslistenprofile auf Warteschlangenmanagerebene oder auf der Ebene der Gruppe mit gemeinsamer Warteschlange in der Klasse MQNLIST oder MXNLIST. Erlauben Sie dann den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile.
   - **Nein**: Definieren Sie ein hlq.NO.NLIST.CHECKS-Profil für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.
8. Benötigen Sie Themensicherheit?
   - **Ja** : Aktivieren Sie die MXTOPIC-Klasse. Definieren Sie geeignete Themenprofile auf Warteschlangenmanagerebene oder auf Ebene der Gruppe mit gemeinsamer Warteschlange in der MXTOPIC-Klasse. Erlauben Sie dann den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile.
   - **Nein**: Definieren Sie ein hlq.NO.TOPIC.CHECKS-Profil für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.
9. Müssen Benutzer die Verwendung der Optionen `MQOPEN` oder `MQPUT1` in Bezug auf die Verwendung von Kontext schützen?
   - **Ja** : Stellen Sie sicher, dass die Klasse MQADMIN oder MXADMIN aktiv ist. Definieren Sie hlq.CONTEXT.queuename-Profile auf Warteschlangenebene, auf Warteschlangenmanagerebene oder auf Ebene der Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN. Erlauben Sie dann den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile.
   - **Nein**: Definieren Sie ein hlq.NO.CONTEXT.CHECKS-Profil für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.
10. Müssen Sie die Verwendung alternativer Benutzer-IDs schützen?
    - **Ja** : Stellen Sie sicher, dass die Klasse MQADMIN oder MXADMIN aktiv ist. Definieren Sie die entsprechenden hlq.ALTERNATE.USER. *alternateuserid*-Profile für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange und ermöglichen Sie den entsprechenden Benutzern oder Gruppen Zugriff auf diese Profile.

- **Nein**: Definieren Sie das Profil hlq.NO.ALTERNATE.USER.CHECKS für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN.

11. Müssen Sie anpassen, welche Benutzer-IDs für die Ressourcensicherheitsüberprüfung über RESLEVEL verwendet werden sollen?
    - **Ja** : Stellen Sie sicher, dass die Klasse MQADMIN oder MXADMIN aktiv ist. Definieren Sie ein hlq.RESLEVEL-Profil auf Warteschlangenmanagerebene oder auf Ebene der Gruppe mit gemeinsamer Warteschlange in der Klasse MQADMIN oder MXADMIN. Erlauben Sie dann den erforderlichen Benutzern oder Gruppen Zugriff auf das Profil.
    - **Nein** : Stellen Sie sicher, dass keine generischen Profile in der Klasse MQADMIN oder MXADMIN vorhanden sind, die auf hlq.RESLEVEL. angewendet werden können. Definieren Sie ein hlq.RESLEVEL-Profil für den erforderlichen Warteschlangenmanager oder die Gruppe mit gemeinsamer Warteschlange und stellen Sie sicher, dass keine Benutzer oder Gruppen darauf zugreifen können.
12. Muss für nicht verwendete Benutzer-IDs aus IBM MQ ein Zeitlimit angegeben werden?
    - **Ja** : Bestimmen Sie, welche Zeitlimitwerte Sie verwenden möchten, und geben Sie den MQSC-Befehl ALTER SECURITY aus, um die Parameter TIMEOUT und INTERVAL zu ändern.
    - **Nein** : Setzen Sie den MQSC-Befehl ALTER SECURITY ab, um den Wert INTERVAL auf null zu setzen.

    Anmerkung: Aktualisieren Sie die von Ihrem Subsystem verwendete Initialisierungseingabedatei CSQINP1, so dass der MQSC-Befehl ALTER SECURITY automatisch ausgegeben wird, wenn der Warteschlangenmanager gestartet wird.
13. Verwenden Sie verteilte Warteschlangensteuerung?
    - **Ja** : Kanalauthentifizierungsdatensätze verwenden. Weitere Informationen finden Sie unter „Kanalauthentifizierungssätze" auf Seite 54.
    - Sie können auch den entsprechenden MCAUSER-Attributwert für jeden Kanal ermitteln oder geeignete Kanalsicherheitsexits bereitstellen.
14. Möchten Sie Transport Layer Security (TLS) verwenden?
    - **Ja** : Um anzugeben, dass jeder Benutzer, der ein TLS-persönliches Zertifikat mit einem angegebenen DN enthält, einen bestimmten MCAUSER verwenden soll, müssen Sie einen Kanalauthentifizierungsdatensatz des Typs SSLPEERMAP festlegen. Sie können einen einzelnen definierten Namen oder ein Muster mit Platzhalterzeichen angeben.
    - Planen Sie Ihre TLS-Infrastruktur. Installieren Sie die System SSL-Funktion von z/OS. Richten Sie in RACF Ihre Zertifikatsnamensfilter (CNFs) ein, falls Sie diese verwenden, sowie Ihre digitalen Zertifikate. Richten Sie Ihren SSL-Schlüsselring ein. Stellen Sie sicher, dass das WS-Manager-Attribut SSLKEYR nicht leer ist und auf Ihren SSL-Schlüsselring verweist. Stellen Sie außerdem sicher, dass der Wert von SSLTASKS mindestens 2 ist.
    - **Nein** : Stellen Sie sicher, dass SSLKEYR leer ist und SSLTASKS null ist.

    Weitere Informationen zu TLS finden Sie unter „TLS-Sicherheitsprotokolle in IBM MQ" auf Seite 26.
15. Verwenden Sie Clients?
    - **Ja** : Kanalauthentifizierungsdatensätze verwenden.
    - Sie können auch den entsprechenden MCAUSER-Attributwert für jeden Serververbindungskanal ermitteln oder bei Bedarf geeignete Kanalsicherheitsexits bereitstellen.
16. Überprüfen Sie die Schaltereinstellungen.

    IBM MQ gibt beim Start des Warteschlangenmanagers Nachrichten aus, in denen Ihre Sicherheitseinstellungen angezeigt werden. Verwenden Sie diese Nachrichten, um festzustellen, ob die Schalter richtig eingestellt sind.
17. Versenden Sie Kennwörter von Clientanwendungen?
    - **Ja**: Stellen Sie für den besten Schutz sicher, dass die Funktion z/OS installiert und Integrated Cryptographic Service Facility (ICSF) gestartet ist.

- **Nein** : Sie können die Fehlermeldung ignorieren, die von ICSF nicht gestartet wurde.

Weitere Informationen zu ICSF finden Sie unter „Integrated Cryptographic Service Facility (ICSF) verwenden” auf Seite 283

Sicherheit konfigurieren

Diese Themensammlung enthält spezifische Informationen zu verschiedenen Betriebssystemen und zur Verwendung von Clients.

ALW Sicherheit unter AIX, Linux, and Windows einrichten

Besondere Sicherheitsaspekte bei Systemen mit AIX, Linux, and Windows

Die Warteschlangenmanager von IBM MQ übertragen meist besonders wichtige Daten. Sie müssen daher ein Berechtigungssystem verwenden, mit dem sichergestellt wird, dass keine unberechtigten Benutzer auf Ihre Warteschlangenmanager zugreifen können. Beachten Sie die folgenden Arten von Sicherheitssteuerungen:

Wer kann IBM MQ verwalten
Sie können die Gruppe der Benutzer definieren, die Befehle für die Verwaltung von IBM MQ ausgeben können.

Wer kann IBM MQ-Objekte verwenden
Sie können definieren, welche Benutzer (in der Regel Anwendungen) MQI-Aufrufe und PCF-Befehle verwenden können, um die folgenden Schritte ausführen zu können:

- Wer kann eine Verbindung zu einem WS-Manager herstellen?
- Wer kann auf Objekte (Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Authentifizierungsinformationsobjekte) zugreifen und welche Art von Zugriff sie auf diese Objekte haben.
- Wer kann auf IBM MQ-Nachrichten zugreifen
- Wer kann auf die Kontextinformationen zugreifen, die einer Nachricht zugeordnet sind.

Kanalsicherheit
Sie müssen sicherstellen, dass Kanäle, die zum Senden von Nachrichten an ferne Systeme verwendet werden, auf die erforderlichen Ressourcen zugreifen können.

Sie können Standardbetriebsfunktionen verwenden, um Zugriff auf Programmbibliotheken, MQI-Linkbibliotheken und Befehle zu erteilen. Das Verzeichnis mit den Warteschlangen und weiteren Warteschlangenmanagerdaten ist allerdings ein nicht öffentliches IBM MQ-Verzeichnis. Verwenden Sie keine Standardbefehle für das Betriebssystem, um Berechtigungen für MQI-Ressourcen zu erteilen oder zu entziehen.

ALW Funktionsweise von Berechtigungen unter AIX, Linux, and Windows

In den Berechtigungsspezifikationstabellen in den Themen in diesem Abschnitt wird genau definiert, wie die Berechtigungen funktionieren, und welche Einschränkungen gelten.

Die Tabellen gelten für die folgenden Situationen:

- Anwendungen, die MQI-Aufrufe absetzen
- Verwaltungsprogramme, die MQSC-Befehle als Escape-PCFs ausgeben
- Verwaltungsprogramme, die PCF-Befehle absetzen

In diesem Abschnitt werden die Informationen in Form einer Gruppe von Tabellen dargestellt, die Folgendes angeben:

Aktion, die ausgeführt werden soll
MQI-Option, MQSC-Befehl oder PCF-Befehl.

Zugriffssteuerungsobjekt
Warteschlange, Prozess, WS-Manager, Namensliste, Authentifizierungsdaten, Kanal, Clientverbindungskanal, Listener oder Service.

Erforderliche Berechtigung
Als MQZAO_-Konstante ausgedrückt.

In den Tabellen entsprechen die von MQZAO_ vorfixierten Konstanten den Schlüsselwörtern in der Berechtigungsliste für den Befehl `setmqaut` für die betreffende Entität. Beispiel: MQZAO_BROWSE entspricht dem Schlüsselwort `+browse`, MQZAO_SET_ALL_CONTEXT entspricht dem Schlüsselwort `+setall` und so weiter. Diese Konstanten werden in der Headerdatei cmqzc.h definiert, die im Lieferumfang des Produkts enthalten ist.

ALW *Berechtigungen für MQI-Aufrufe*

Für **MQCONN** , **MQOPEN** , **MQPUT1** und **MQCLOSE** sind möglicherweise Berechtigungsprüfungen erforderlich. In den Tabellen in diesem Thema werden die Berechtigungen zusammengefasst, die für die einzelnen Telefonanrufe benötigt werden.

Eine Anwendung darf bestimmte MQI-Aufrufe und -Optionen nur dann absetzen, wenn die Benutzer-ID, unter der sie ausgeführt wird (oder deren Berechtigungen vorausgesetzt werden können), die entsprechende Berechtigung erteilt hat.

Für vier MQI-Aufrufe sind möglicherweise Berechtigungsprüfungen erforderlich: **MQCONN** , **MQOPEN** , **MQPUT1** und **MQCLOSE** .

Für **MQOPEN** und **MQPUT1** wird die Berechtigungs-Prüfung auf den Namen des zu öffnende Objekts und nicht auf den Namen oder Namen durchgeführt, die sich nach dem Namen eines Namens ergeben. Beispielsweise kann einer Anwendung die Berechtigung zum Öffnen einer Aliaswarteschlange erteilt werden, ohne die Berechtigung zum Öffnen der Basiswarteschlange, in die der Aliasname aufgelöst wird. Die Regel ist, dass die Prüfung bei der ersten Definition ausgeführt wird, die während des Prozesses zur Auflösung eines Namens gefunden wird, der kein WS-Manager-Aliasname ist, es sei denn, die Definition des WS-Manager-Aliasnamens wird direkt geöffnet. Das heißt, sein Name wird im Feld *ObjectName* des Objektdeskriptors angezeigt. Die Berechtigung wird immer für das Objekt benötigt, das geöffnet wird. In einigen Fällen ist eine zusätzliche warteschlangenunabhängige Berechtigung erforderlich, die über eine Berechtigung für das WS-Manager-Objekt ermittelt wird.

In Tabelle 10 auf Seite 142, Tabelle 11 auf Seite 143, Tabelle 12 auf Seite 143 und Tabelle 13 auf Seite 144 sind die für die einzelnen Aufrufe erforderlichen Berechtigungen zusammengestellt. In den Tabellen bedeutet *Nicht zutreffend* , dass die Berechtigungsprüfung für diese Operation nicht relevant ist. *Kein Prüfvorgang* bedeutet, dass keine Berechtigungsprüfung ausgeführt wird.

Anmerkung: In diesen Tabellen finden Sie keine Erwähnung von Namenslisten, Kanälen, Clientverbindungskanälen, Empfangsprogrammen, Services oder Authentifizierungsinformationsobjekten. Dies liegt daran, dass keine der Berechtigungen für diese Objekte gilt, mit Ausnahme von MQOO_INQUIRE, für die die gleichen Berechtigungen wie für die anderen Objekte gelten.

Die Sonderberechtigung MQZAO_ALL_MQI enthält alle Berechtigungen in den Tabellen, die für den Objekttyp relevant sind, mit Ausnahme von MQZAO_DELETE und MQZAO_DISPLAY, die als Verwaltungsberechtigungen klassifiziert werden.

Wenn Sie die Optionen für den Nachrichtenkontext ändern möchten, müssen Sie über die entsprechenden Berechtigungen zum Aufrufen des Aufrufs verfügen. Zur Ausführung von MQOO_SET_IDENTITY_CONTEXT oder MQPMO_SET_IDENTITY_CONTEXT benötigen Sie zum Beispiel die Berechtigung `+setid`.

Tabelle 10. Für MQCONN-Aufrufe erforderliche Sicherheitsberechtigung

Erforderliche Berechtigung für:	Warteschlangenobjekt („1” auf Seite 144)	Prozessobjekt	WS-Manager-Objekt
MQCONN	-	-	MQZAO_CONNECT

Tabelle 11. Für MQOPEN-Aufrufe erforderliche Sicherheitsberechtigung			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 144)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQOO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE
MQOO_BROWSE	MQZAO_BROWSE	-	Keine Prüfung
MQOO_INPUT_*	MQZAO_INPUT	-	Keine Prüfung
MQOO_SAVE_ ALL_CONTEXT **(„2” auf Seite 144)**	MQZAO_INPUT	-	-
MQOO_OUTPUT (normale Warteschlange) **(„3” auf Seite 144)**	MQZAO_OUTPUT	-	-
MQOO_PASS_ IDENTITY_CONTEXT **(„4” auf Seite 144)**	MQZAO_PASS_ IDENTITY_CONTEXT	-	Keine Prüfung
MQOO_PASS_ALL_ CONTEXT **(„4” auf Seite 144, „5” auf Seite 144)**	MQZAO_PASS _ALL_CONTEXT	-	Keine Prüfung
MQOO_SET_ IDENTITY_CONTEXT **(„4” auf Seite 144, „5” auf Seite 144)**	MQZAO_SET_ IDENTITY_CONTEXT	-	MQZAO_SET_ IDENTITY_CONTEXT **(„6” auf Seite 144)**
MQOO_SET_ ALL_CONTEXT **(„4” auf Seite 144, „7” auf Seite 144)**	MQZAO_SET_ ALL_CONTEXT	-	MQZAO_SET_ ALL_CONTEXT **(„6” auf Seite 144)**
MQOO_OUTPUT (Übertragungswarteschlange) **(„8” auf Seite 144)**	MQZAO_SET_ ALL_CONTEXT	-	MQZAO_SET_ ALL_CONTEXT **(„6” auf Seite 144)**
MQOO_SET	MQZAO_SET	-	Keine Prüfung
MQOO_ALTERNATE_ USER_AUTHORITY	**(„9” auf Seite 144)**	**(„9” auf Seite 144)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(„9” auf Seite 144, „10” auf Seite 145)**

Tabelle 12. Für MQPUT1-Aufrufe erforderliche Sicherheitsberechtigung			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 144)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(„11” auf Seite 145)**	-	Keine Prüfung
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(„11” auf Seite 145)**	-	Keine Prüfung
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(„11” auf Seite 145)**	-	MQZAO_SET_ IDENTITY_CONTEXT **(„6” auf Seite 144)**

Tabelle 12. Für MQPUT1-Aufrufe erforderliche Sicherheitsberechtigung (Forts.)			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 144)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(„11” auf Seite 145)**	-	MQZAO_SET_ ALL_CONTEXT **(„6” auf Seite 144)**
(Übertragungswarteschlange) **(„8” auf Seite 144)**	MQZAO_SET_ ALL_CONTEXT	-	MQZAO_SET_ ALL_CONTEXT **(„6” auf Seite 144)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(„12” auf Seite 145)**	-	MQZAO_ALTERNATE_ USER_AUTHORITY **(„10” auf Seite 145)**

Tabelle 13. Für MQCLOSE-Aufrufe erforderliche Sicherheitsberechtigung			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 144)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQCO_DELETE	MQZAO_DELETE **(„13” auf Seite 145)**	-	-
MQCO_DELETE _PURGE	MQZAO_DELETE **(„13” auf Seite 145)**	-	-

Hinweise zu den Tabellen:

1. Beim Öffnen einer Modellwarteschlange:
   - Die Berechtigung MQZAO_DISPLAY wird für die Modellwarteschlange zusätzlich zur Berechtigung zum Öffnen der Modellwarteschlange für den Typ des Zugriffs, für den Sie geöffnet werden, benötigt.
   - Die Berechtigung MQZAO_CREATE ist nicht erforderlich, um die dynamische Warteschlange zu erstellen.
   - Die Benutzer-ID, die zum Öffnen der Modellwarteschlange verwendet wird, wird automatisch allen warteschlangenspezifischen Berechtigungen (äquivalent zu MQZAO_ALL) für die erstellte dynamische Warteschlange erteilt.
2. MQOO_INPUT_ * muss ebenfalls angegeben werden. Dies gilt für eine lokale, eine Modell-oder eine Aliaswarteschlange.
3. Diese Prüfung wird für alle ausgehenden Fälle, außer für Übertragungswarteschlangen ausgeführt (siehe Anmerkung „8” auf Seite 144).
4. MQOO_OUTPUT muss ebenfalls angegeben werden.
5. MQOO_PASS_IDENTITY_CONTEXT wird auch von dieser Option impliziert.
6. Diese Berechtigung ist sowohl für das Warteschlangenmanagerobjekt als auch für die bestimmte Warteschlange erforderlich.
7. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT und MQOO_SET_IDENTITY_CONTEXT werden ebenfalls von dieser Option impliziert.
8. Diese Prüfung wird für eine lokale oder Modellwarteschlange ausgeführt, die über ein *Usage* -Warteschlangenattribut von MQUS_TRANSMISSION verfügt und direkt für die Ausgabe geöffnet wird. Sie findet keine Anwendung, wenn eine ferne Warteschlange geöffnet wird (entweder durch Angabe der Namen des fernen Warteschlangenmanagers und der fernen Warteschlange oder durch Angabe des Namens einer lokalen Definition der fernen Warteschlange).
9. Es muss auch mindestens ein MQOO_INQUIRE (für einen beliebigen Objekttyp) oder MQOO_BROWSE, MQOO_INPUT_ *, MQOO_OUTPUT oder MQOO_SET (für Warteschlangen) angegeben werden.

Die durchgeführte Prüfung erfolgt wie bei den anderen angegebenen Optionen unter Verwendung der angegebenen alternativen Benutzer-ID für die spezielle Objektberechtigung und der aktuellen Anwendungsberechtigung für die Prüfung MQZAO_ALTERNATE_USER_IDENTIFIER.

10. Mit dieser Berechtigung kann jede beliebige *AlternateUserId* angegeben werden.
11. Es wird auch eine MQZAO_OUTPUT-Prüfung durchgeführt, wenn die Warteschlange kein Warteschlangenattribut *Usage* von MQUS_TRANSMISSION hat.
12. Die durchgeführte Prüfung erfolgt wie bei den anderen angegebenen Optionen unter Verwendung der angegebenen alternativen Benutzer-ID für die benannte Warteschlangenberechtigung und der aktuellen Anwendungsberechtigung für die Prüfung MQZAO_ALTERNATE_USER_IDENTIFIER.
13. Die Prüfung wird nur durchgeführt, wenn beide der folgenden Aussagen wahr sind:
    - Eine permanente dynamische Warteschlange wird geschlossen und gelöscht.
    - Die Warteschlange wurde nicht durch den Aufruf `MQOPEN` erstellt, der die verwendete Objektkennung zurückgegeben hat.

    Sonst gibt es keine Prüfung.

ALW *Berechtigungen für MQSC-Befehle in Escape-PCFs*

In diesen Informationen werden die Berechtigungen zusammengefasst, die für jeden in Escape PCF enthaltenen MQSC-Befehl erforderlich sind.

Nicht zutreffend bedeutet, dass diese Operation für diesen Objekttyp nicht relevant ist.

Die Benutzer-ID, unter der das Programm, das den Befehl übergibt, ausgeführt wird, muss außerdem über die folgenden Berechtigungen verfügen:

- Berechtigung MQZAO_CONNECT für den WS-Manager
- MQZAO_DISPLAY-Berechtigung auf dem Warteschlangenmanager, um PCF-Befehle auszuführen
- Berechtigung zum Absetzen des MQSC-Befehls im Text des Escape-PCF-Befehls

ALTER *object*

Objekt	**Erforderliche Berechtigung**
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	MQZAO_ÄNDERUNG
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsdaten	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Empfangsprogramm	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG
Kommunikationsinformationen	MQZAO_ÄNDERUNG

CLEAR *object*

Objekt	**Erforderliche Berechtigung**
Warteschlange	MQZAO_CLEAR
Thema	MQZAO_CLEAR

Objekt	Erforderliche Berechtigung
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	Nicht zutreffend
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend
Kommunikationsinformationen	Nicht zutreffend

DEFINE *Objekt* NOREPLACE („1" auf Seite 150)

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(„2" auf Seite 150)**
Thema	MQZAO_CREATE **(„2" auf Seite 150)**
Prozess	MQZAO_CREATE **(„2" auf Seite 150)**
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_CREATE **(„2" auf Seite 150)**
Authentifizierungsdaten	MQZAO_CREATE **(„2" auf Seite 150)**
Kanal	MQZAO_CREATE **(„2" auf Seite 150)**
Clientverbindungskanal	MQZAO_CREATE **(„2" auf Seite 150)**
Empfangsprogramm	MQZAO_CREATE **(„2" auf Seite 150)**
Service	MQZAO_CREATE **(„2" auf Seite 150)**
Kommunikationsinformationen	MQZAO_CREATE **(„2" auf Seite 150)**

DEFINE *object* REPLACE („1" auf Seite 150, „3" auf Seite 150)

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsdaten	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Empfangsprogramm	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG

Objekt	Erforderliche Berechtigung
Kommunikationsinformationen	MQZAO_ÄNDERUNG

DELETE *object*

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DELETE
Thema	MQZAO_DELETE
Prozess	MQZAO_DELETE
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_DELETE
Authentifizierungsdaten	MQZAO_DELETE
Kanal	MQZAO_DELETE
Clientverbindungskanal	MQZAO_DELETE
Empfangsprogramm	MQZAO_DELETE
Service	MQZAO_DELETE
Kommunikationsinformationen	MQZAO_DELETE

DISPLAY *object*

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DISPLAY
Thema	MQZAO_DISPLAY
Prozess	MQZAO_DISPLAY
Warteschlangenmanager	MQZAO_DISPLAY
Namensliste	MQZAO_DISPLAY
Authentifizierungsdaten	MQZAO_DISPLAY
Kanal	MQZAO_DISPLAY
Clientverbindungskanal	MQZAO_DISPLAY
Empfangsprogramm	MQZAO_DISPLAY
Service	MQZAO_DISPLAY
Kommunikationsinformationen	MQZAO_DISPLAY

START *object*

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend

Objekt	Erforderliche Berechtigung
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	MQZAO_CONTROL
Service	MQZAO_CONTROL
Kommunikationsinformationen	Nicht zutreffend

STOP *object*

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	MQZAO_CONTROL
Service	MQZAO_CONTROL
Kommunikationsinformationen	Nicht zutreffend

Kanalbefehle

Befehl	Objekt	Erforderliche Berechtigung
PING CHANNEL	Kanal	MQZAO_CONTROL
RESET CHANNEL	Kanal	MQZAO_CONTROL_EXTENDED
GELÖST-CHANNEL	Kanal	MQZAO_CONTROL_EXTENDED

Subskriptionsbefehle

Befehl	Objekt	Erforderliche Berechtigung
ALTER SUB	Thema	MQZAO_CONTROL
SUB DEFINI	Thema	MQZAO_CONTROL
DELETE SUB	Thema	MQZAO_CONTROL
ANZEIGEN SUB	Thema	MQZAO_DISPLAY

Sicherheitsbefehle

Befehl	Objekt	Erforderliche Berechtigung
SET AUTHREC	Warteschlangenmanager	MQZAO_ÄNDERUNG

Befehl	Objekt	Erforderliche Berechtigung
AUTHREC löschen	Warteschlangenmanager	MQZAO_ÄNDERUNG
ANZEIGEN AUTHREC	Warteschlangenmanager	MQZAO_DISPLAY
ANZEIGEN AUTHSERV	Warteschlangenmanager	MQZAO_DISPLAY
ANZEIGEN ENTAUTH	Warteschlangenmanager	MQZAO_DISPLAY
SET CHLAUTH	Warteschlangenmanager	MQZAO_ÄNDERUNG
ANZEIGEN CHLAUTH	Warteschlangenmanager	MQZAO_DISPLAY
REFRESH SECURITY	Warteschlangenmanager	MQZAO_ÄNDERUNG

Statusanzeigen

Befehl	Objekt	Erforderliche Berechtigung
ANZEIGEN CHSTATUS	Warteschlangenmanager	MQZAO_DISPLAY Beachten Sie, dass die Berechtigung +inq (oder äquivalent MQZAO_INQUIRE) in der Übertragungswarteschlange erforderlich ist, wenn der Kanaltyp CLUSSDR ist.
ANZEIGEN LSSTATUS	Warteschlangenmanager	MQZAO_DISPLAY
DISPLAY PUBSUB	Warteschlangenmanager	MQZAO_DISPLAY
ANZEIGEN SBSTATUS	Warteschlangenmanager	MQZAO_DISPLAY
ANZEIGEN SVSTATUS	Warteschlangenmanager	MQZAO_DISPLAY
ANZEIGEN TPSTATUS	Warteschlangenmanager	MQZAO_DISPLAY

Clusterbefehle

Befehl	Objekt	Erforderliche Berechtigung
DISPLAY CLUSQMGR	Warteschlangenmanager	MQZAO_DISPLAY
REFRESH CLUSTER	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	
RESET CLUSTER	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	
SUSPEND QMGR	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	
RESUME QMGR	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	

Andere Verwaltungsbefehle

Befehl	Objekt	Erforderliche Berechtigung
PING QMGR	Warteschlangenmanager	MQZAO_DISPLAY
REFRESH QMGR	Warteschlangenmanager	MQZAO_ÄNDERUNG
RESET QMGR	Warteschlangenmanager	MQZAO_ÄNDERUNG
DISPLAY CONN	Warteschlangenmanager	MQZAO_DISPLAY
STOP CONN	Warteschlangenmanager	MQZAO_ÄNDERUNG

Anmerkung:

1. Bei DEFINE-Befehlen wird die Berechtigung MQZAO_DISPLAY auch für das LIKE-Objekt benötigt, wenn ein Objekt angegeben wird, oder auf dem entsprechenden Objekt SYSTEM.DEFAULT.xxx, wenn LIKE weggelassen wird.
2. Die Berechtigung MQZAO_CREATE ist nicht spezifisch für ein bestimmtes Objekt oder einen bestimmten Objekttyp. Die Erstellungsberechtigung wird für alle Objekte für einen bestimmten Warteschlangenmanager erteilt, indem ein Objekttyp QMGR im Befehl `setmqaut` angegeben wird.
3. Dies gilt, wenn das zu ersetzendes Objekt bereits vorhanden ist. Ist dies nicht der Fall, ist die Prüfung wie für DEFINE *object* NOREPLACE.

Zugehörige Informationen

Clustering: Best Practices für REFRESH CLUSTER verwenden

ALW *Berechtigungen für PCF-Befehle*

In diesem Abschnitt werden die Berechtigungen zusammengefasst, die für die einzelnen PCF-Befehle erforderlich sind.

Keine Prüfung bedeutet, dass keine Berechtigungsprüfung durchgeführt wird; *Nicht zutreffend* bedeutet, dass diese Operation für diesen Objekttyp nicht relevant ist.

Die Benutzer-ID, unter der das Programm, das den Befehl übergibt, ausgeführt wird, muss außerdem über die folgenden Berechtigungen verfügen:

- Berechtigung MQZAO_CONNECT für den WS-Manager
- MQZAO_DISPLAY-Berechtigung auf dem Warteschlangenmanager, um PCF-Befehle auszuführen

Die Sonderberechtigung MQZAO_ALL_ADMIN enthält alle Berechtigungen in der folgenden Liste, die für den Objekttyp relevant sind, mit Ausnahme von MQZAO_CREATE, die nicht spezifisch für ein bestimmtes Objekt oder einen bestimmten Objekttyp ist.

Change *object*

Object	**Erforderliche Berechtigung**
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
WS-Manager	MQZAO_ÄNDERUNG
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsinformationen	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Listener	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG
Kommunikationsinformationen	MQZAO_ÄNDERUNG

Löschen Sie *object*.

Object	**Erforderliche Berechtigung**
Warteschlange	MQZAO_CLEAR
Thema	MQZAO_CLEAR

Object	Erforderliche Berechtigung
Prozess	-
Warteschlangenmanager	-
Namensliste	-
Authentifizierungsinformationen	-
Kanal	-
Clientverbindungskanal	-
Empfangsprogramm	-
Service	-
Kommunikationsinformationen	-

Kopieren Sie *object* (ohne Ersetzen) (1)

Object	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(2)**
Thema	MQZAO_CREATE **(2)**
Prozess	MQZAO_CREATE **(2)**
Warteschlangenmanager	-
Namensliste	MQZAO_CREATE **(2)**
Authentifizierungsinformationen	MQZAO_CREATE **(2)**
Kanal	MQZAO_CREATE **(2)**
Clientverbindungskanal	MQZAO_CREATE **(2)**
Listener	MQZAO_CREATE **(2)**
Service	MQZAO_CREATE **(2)**
Kommunikationsinformationen	MQZAO_CREATE **(„2” auf Seite 156)**

Kopieren *object* (mit Ersetzen) (1 , 4)

Object	Erforderliche Berechtigung
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	-
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsinformationen	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Listener	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG

Object	Erforderliche Berechtigung
Kommunikationsinformationen	MQZAO_ÄNDERUNG

Erstellen Sie *object* (ohne Ersetzen) (3)

Object	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(2)**
Thema	MQZAO_CREATE **(2)**
Prozess	MQZAO_CREATE **(2)**
Warteschlangenmanager	-
Namensliste	MQZAO_CREATE **(2)**
Authentifizierungsinformationen	MQZAO_CREATE **(2)**
Kanal	MQZAO_CREATE **(2)**
Clientverbindungskanal	MQZAO_CREATE **(2)**
Listener	MQZAO_CREATE **(2)**
Service	MQZAO_CREATE **(2)**
Kommunikationsinformationen	MQZAO_CREATE **(2)**

Erstellen Sie *object* (mit Ersetzen) (3 , 4)

Object	Erforderliche Berechtigung
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	-
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsinformationen	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Listener	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG
Kommunikationsinformationen	MQZAO_ÄNDERUNG

***object* löschen**

Object	Erforderliche Berechtigung
Warteschlange	MQZAO_DELETE
Thema	MQZAO_DELETE
Prozess	MQZAO_DELETE
Warteschlangenmanager	-
Namensliste	MQZAO_DELETE

Object	Erforderliche Berechtigung
Authentifizierungsinformationen	MQZAO_DELETE
Kanal	MQZAO_DELETE
Clientverbindungskanal	MQZAO_DELETE
Listener	MQZAO_DELETE
Service	MQZAO_DELETE
Kommunikationsinformationen	MQZAO_DELETE

Inquire *object*

Object	Erforderliche Berechtigung
Warteschlange	MQZAO_DISPLAY
Thema	MQZAO_DISPLAY
Prozess	MQZAO_DISPLAY
WS-Manager	MQZAO_DISPLAY
Namensliste	MQZAO_DISPLAY
Authentifizierungsinformationen	MQZAO_DISPLAY
Kanal	MQZAO_DISPLAY
Clientverbindungskanal	MQZAO_DISPLAY
Listener	MQZAO_DISPLAY
Service	MQZAO_DISPLAY
Kommunikationsinformationen	MQZAO_DISPLAY

***object* -Namen inquire**

Object	Erforderliche Berechtigung
Warteschlange	Keine Prüfung
Thema	Keine Prüfung
Prozess	Keine Prüfung
Warteschlangenmanager	Keine Prüfung
Namensliste	Keine Prüfung
Authentifizierungsinformationen	Keine Prüfung
Kanal	Keine Prüfung
Clientverbindungskanal	Keine Prüfung
Empfangsprogramm	Keine Prüfung
Service	Keine Prüfung
Kommunikationsinformationen	Keine Prüfung

***object* starten**

Object	Erforderliche Berechtigung
Warteschlange	-
Thema	-
Prozess	-
Warteschlangenmanager	-
Namensliste	-
Authentifizierungsinformationen	-
Kanal	MQZAO_CONTROL
Clientverbindungskanal	-
Listener	MQZAO_CONTROL
Service	MQZAO_CONTROL
Kommunikationsinformationen	-

***object* stoppen**

Object	Erforderliche Berechtigung
Warteschlange	-
Thema	-
Prozess	-
Warteschlangenmanager	-
Namensliste	-
Authentifizierungsinformationen	-
Kanal	MQZAO_CONTROL
Clientverbindungskanal	-
Listener	MQZAO_CONTROL
Service	MQZAO_CONTROL
Kommunikationsinformationen	-

Kanalbefehle

Befehl	Object	Erforderliche Berechtigung
Pingkanal	Kanal	MQZAO_CONTROL
Kanal zurücksetzen	Kanal	MQZAO_CONTROL_EXTENDED
Auflösungskanal	Kanal	MQZAO_CONTROL_EXTENDED

Subskriptionsbefehle

Befehl	Object	Erforderliche Berechtigung
Subskription ändern	Thema	MQZAO_CONTROL
Subskription erstellen	Thema	MQZAO_CONTROL

Befehl	Object	Erforderliche Berechtigung
Subskription löschen	Thema	MQZAO_CONTROL
Inquire Subscription	Thema	MQZAO_DISPLAY

Sicherheitsbefehle

Befehl	Object	Erforderliche Berechtigung
Berechtigungssatz festlegen	Warteschlangenmanager	MQZAO_ÄNDERUNG
Berechtigungsdatensatz löschen	Warteschlangenmanager	MQZAO_ÄNDERUNG
Berechtigungsdatensätze anfragen	Warteschlangenmanager	MQZAO_DISPLAY
Inquire Authority Service	Warteschlangenmanager	MQZAO_DISPLAY
Inquire Entity Authority	Warteschlangenmanager	MQZAO_DISPLAY
Kanalauthentifizierungsdatensatz festlegen	Warteschlangenmanager	MQZAO_ÄNDERUNG
Kanalauthentifizierungsdatensätze abgefragt	Warteschlangenmanager	MQZAO_DISPLAY
Sicherheit aktualisieren	Warteschlangenmanager	MQZAO_ÄNDERUNG

Statusanzeigen

Befehl	Object	Erforderliche Berechtigung
Inquire Channel Status	Warteschlangenmanager	MQZAO_DISPLAY Beachten Sie, dass die Berechtigung `+inq` (oder äquivalent MQZAO_INQUIRE) in der Übertragungswarteschlange erforderlich ist, wenn der Kanaltyp CLUSSDR ist.
Status des Inquire-Channel-Listeners	Warteschlangenmanager	MQZAO_DISPLAY
Publish/Subscribe-Status von 'Inquire'	Warteschlangenmanager	MQZAO_DISPLAY
Subskriptionsstatus der Inquire-Funktion	Warteschlangenmanager	MQZAO_DISPLAY
Status des Service 'Inquire'	Warteschlangenmanager	MQZAO_DISPLAY
Inquire-Themenstatus	Warteschlangenmanager	MQZAO_DISPLAY

Clusterbefehle

Befehl	Object	Erforderliche Berechtigung
Clusterwarteschlangenmanager anfragen	Warteschlangenmanager	MQZAO_DISPLAY
Cluster aktualisieren	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich

Befehl	Object	Erforderliche Berechtigung
Cluster zurücksetzen	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich
Clusterwarteschlangenmanager-Cluster aussetzen	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich
WS-Manager-Cluster wieder aufnehmen	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich

Andere Verwaltungsbefehle

Befehl	Object	Erforderliche Berechtigung
Ping-WS-Manager	Warteschlangenmanager	MQZAO_DISPLAY
Warteschlangenmanager aktualisieren	Warteschlangenmanager	MQZAO_ÄNDERUNG
Warteschlangenmanager zurücksetzen	Warteschlangenmanager	MQZAO_ÄNDERUNG
Warteschlangenstatistik zurücksetzen	Warteschlange	MQZAO_DISPLAY und MQZAO_CHANGE
Verbindungsanfragung	Warteschlangenmanager	MQZAO_DISPLAY
Verbindung stoppen	Warteschlangenmanager	MQZAO_ÄNDERUNG

Anmerkung:

1. Für Kopierbefehle ist auch die Berechtigung MQZAO_DISPLAY für das From-Objekt erforderlich.
2. Die Berechtigung MQZAO_CREATE ist nicht spezifisch für ein bestimmtes Objekt oder einen bestimmten Objekttyp. Die Erstellungsberechtigung wird für alle Objekte für einen bestimmten Warteschlangenmanager erteilt, indem ein Objekttyp QMGR im Befehl `setmqaut` angegeben wird.
3. Bei Erstellungsbefehlen wird die Berechtigung MQZAO_DISPLAY auch für das entsprechende Objekt SYSTEM.DEFAULT.* benötigt.
4. Dies gilt, wenn das zu ersetzendes Objekt bereits vorhanden ist. Ist dies nicht der Fall, ist die Prüfung wie für Kopieren oder Erstellen ohne Ersetzen.

AIX Gruppen unter AIX erstellen und verwalten

Wenn Sie in AIX nicht mit NIS oder NIS+ arbeiten, verwenden Sie SMITTY für die Arbeit mit Gruppen.

Informationen zu diesem Vorgang

Unter AIX können Sie mit SMITTY eine Gruppe erstellen, einen Benutzer zu einer Gruppe hinzufügen, eine Liste der Benutzer in der Gruppe anzeigen und einen Benutzer aus einer Gruppe entfernen.

Vorgehensweise

1. Wählen Sie in SMITTY die Option **Security and Users** (Sicherheit und Benutzer) aus und drücken Sie die Eingabetaste.
2. Wählen Sie **Groups** (Gruppen) aus und drücken Sie die Eingabetaste.
3. Führen Sie die folgenden Schritte aus, um eine Gruppe zu erstellen:
   a) Wählen Sie **Add a Group** (Gruppe hinzufügen) aus und drücken Sie die Eingabetaste.
   b) Geben Sie den Namen der Gruppe und die Namen der Benutzer ein, die der Gruppe hinzugefügt werden sollen, getrennt durch Kommas.

c) Drücken Sie die Eingabetaste, um die Gruppe zu erstellen.

4. Führen Sie zum Hinzufügen eines Benutzers zu einer Gruppe die folgenden Schritte aus:
   a) Wählen Sie **Change / Show Characteristics of Groups** (Merkmale von Gruppen ändern/anzeigen) und drücken Sie die Eingabetaste.
   b) Geben Sie den Namen der Gruppe ein, um eine Liste der Mitglieder der Gruppe anzuzeigen.
   c) Fügen Sie die Namen der Benutzer, die der Gruppe hinzugefügt werden sollen, durch Kommas getrennt hinzu.
   d) Drücken Sie die Eingabetaste, um die Namen der Gruppe hinzuzufügen.
5. Führen Sie die folgenden Schritte aus, um anzuzeigen, wer sich in einer Gruppe befindet:
   a) Wählen Sie **Change / Show Characteristics of Groups** (Merkmale von Gruppen ändern/anzeigen) und drücken Sie die Eingabetaste.
   b) Geben Sie den Namen der Gruppe ein, um eine Liste der Mitglieder der Gruppe anzuzeigen.
6. Führen Sie zum Entfernen eines Benutzers aus einer Gruppe die folgenden Schritte aus:
   a) Wählen Sie **Change / Show Characteristics of Groups** (Merkmale von Gruppen ändern/anzeigen) und drücken Sie die Eingabetaste.
   b) Geben Sie den Namen der Gruppe ein, um eine Liste der Mitglieder der Gruppe anzuzeigen.
   c) Löschen Sie den Namen des Benutzers, der aus der Gruppe entfernt werden soll.
   d) Drücken Sie die Eingabetaste, um den Namen aus der Gruppe zu entfernen.

Linux Gruppen unter Linux erstellen und verwalten

Wenn Sie in Linux nicht NIS oder NIS+ verwenden, verwenden Sie zur Arbeit mit Gruppen die Datei `/etc/group`.

Informationen zu diesem Vorgang

Unter Linux werden Gruppeninformationen in der Datei `/etc/group` gespeichert. Mit Befehlen können Sie eine Gruppe erstellen, einen Benutzer zu einer Gruppe hinzufügen, eine Liste der Benutzer in der Gruppe anzeigen und einen Benutzer aus einer Gruppe entfernen.

Vorgehensweise

1. Zum Erstellen einer neuen Gruppe verwenden Sie den Befehl **groupadd**.
   Geben Sie den folgenden Befehl ein:

   ```
   groupadd -g group-ID group-name
   ```

   Dabei ist *Gruppen-ID* die numerische ID der Gruppe und *Gruppenname* ist der Name der Gruppe.
2. Wenn Sie einer ergänzenden Gruppe ein Mitglied hinzufügen möchten, führen Sie mit dem Befehl **usermod** die ergänzenden Gruppen auf, in denen der Benutzer aktuell Mitglied ist, sowie die ergänzenden Gruppen, zu denen der Benutzer gehören soll.
   Wenn der Benutzer beispielsweise bereits Mitglied der Gruppe `groupa` ist und der Gruppe `groupb` zugeordnet werden soll, verwenden Sie den folgenden Befehl:

   ```
   usermod -G groupa,groupb user-name
   ```

   Dabei ist *Benutzername* der Name des Benutzers.
3. Mit dem Befehl **getent** können Sie die Mitglieder einer Gruppe anzeigen.
   Geben Sie den folgenden Befehl ein:

   ```
   getent group group-name
   ```

   Dabei steht *Gruppenname* für den Namen der Gruppe.

4. Wenn Sie ein Mitglied aus einer ergänzenden Gruppe entfernen möchten, verwenden Sie den Befehl **usermod**, um die ergänzenden Gruppen aufzuführen, in denen der Benutzer weiterhin Mitglied sein soll.
   Wenn die Primärgruppe des Benutzers beispielsweise `users` ist und der Benutzer außerdem Mitglied der Gruppen `mqm`, `groupa` und `groupb` ist, verwenden Sie zum Entfernen des Benutzers aus der Gruppe `mqm` den folgenden Befehl:

   ```
   usermod -G groupa,groupb user-name
   ```

   Dabei ist *Benutzername* der Name des Benutzers.

Windows Gruppen unter Windows erstellen und verwalten

Verwenden Sie unter Windows die Funktion 'Computerverwaltung', um Gruppen auf einer Workstation oder einem Mitgliedsserver zu verwalten.

Informationen zu diesem Vorgang

Für Domänencontroller werden Benutzer und Gruppen über Active Directory verwaltet. Weitere Informationen zur Verwendung von Active Directory finden Sie in den entsprechenden Betriebssystemanweisungen.

Alle Änderungen, die Sie an der Gruppenzugehörigkeit eines Prinzipals vornehmen, werden erst erkannt, wenn der Warteschlangenmanager erneut gestartet wird oder Sie den MQSC-Befehl **REFRESH SECURITY** (oder die PCF-Entsprechung) ausgeben.

Verwenden Sie die Windows-Anzeige 'Computerverwaltung', um mit Benutzern und Gruppen zu arbeiten. Alle Änderungen, die an dem aktuellen angemeldeten Benutzer vorgenommen wurden, sind möglicherweise erst wirksam, wenn sich der Benutzer erneut anmeldet.

Windows *Gruppe unter Windows erstellen*

Erstellen Sie eine Gruppe, indem Sie die Steuerkonsole verwenden.

Vorgehensweise

1. Steuerkonsole öffnen
2. Klicken Sie doppelt auf **Verwaltung** .
   Die Anzeige mit den Verwaltungstools wird geöffnet.
3. Klicken Sie doppelt auf **Computerverwaltung** .
   Die Anzeige 'Computerverwaltung' wird geöffnet.
4. Erweitern Sie **Lokale Benutzer und Gruppen** .
5. Klicken Sie auf **Gruppen** , und wählen Sie **Neue Gruppe ...** aus.
   Das Fenster 'Neue Gruppe' wird angezeigt.
6. Geben Sie einen geeigneten Namen in das Feld Gruppenname ein, und klicken Sie anschließend auf **Erstellen** .
7. Klicken Sie auf **Schließen** .

Windows *Benutzer unter Windows einer Gruppe hinzufügen*

Fügen Sie einen Benutzer mithilfe der Steuerkonsole zu einer Gruppe hinzu.

Vorgehensweise

1. Steuerkonsole öffnen
2. Klicken Sie doppelt auf **Verwaltung** .
   Die Anzeige mit den Verwaltungstools wird geöffnet.
3. Klicken Sie doppelt auf **Computerverwaltung** .

Die Anzeige 'Computerverwaltung' wird geöffnet.

4. Erweitern Sie in der Anzeige 'Computerverwaltung' den Eintrag **Lokale Benutzer und Gruppen** .
5. Wählen Sie **Benutzer** aus.
6. Klicken Sie doppelt auf den Benutzer, der zu einer Gruppe hinzugefügt werden soll.
   Die Anzeige mit den Benutzereigenschaften wird angezeigt.
7. Wählen Sie die Registerkarte **Mitglied von** aus.
8. Wählen Sie die Gruppe aus, der der Benutzer hinzugefügt werden soll. Wenn die gewünschte Gruppe nicht sichtbar ist:
   a) Klicken Sie auf **Hinzufügen ...**.
      Daraufhin wird die Anzeige "Gruppen auswählen" aufgerufen.
   b) Klicken Sie auf **Positionen ...** .
      Die Anzeige "Standorte" wird angezeigt.
   c) Wählen Sie in der Liste die Position der Gruppe aus, der Sie den Benutzer hinzufügen möchten, und klicken Sie auf **OK** .
   d) Geben Sie den Gruppennamen in das angegebene Feld ein.

      Klicken Sie alternativ auf **Erweitert ...** . und dann **Jetzt suchen** , um die Gruppen aufzulisten, die an der aktuell ausgewählten Position verfügbar sind. Wählen Sie in dieser Gruppe die Gruppe aus, der Sie den Benutzer hinzufügen möchten, und klicken Sie auf **OK** .
   e) Klicken Sie auf **OK**.
      Die Anzeige mit den Benutzereigenschaften wird angezeigt, in der die hinzugefügte Gruppe angezeigt wird.
   f) Wählen Sie die Gruppe aus.
9. Klicken Sie auf **OK**.
   Die Anzeige 'Computerverwaltung' wird angezeigt.

Windows *Mitglieder in einer Gruppe unter Windows anzeigen*

Zeigen Sie die Mitglieder einer Gruppe an, indem Sie die Steuerkonsole verwenden.

Vorgehensweise

1. Steuerkonsole öffnen
2. Klicken Sie doppelt auf **Verwaltung** .
   Die Anzeige mit den Verwaltungstools wird geöffnet.
3. Klicken Sie doppelt auf **Computerverwaltung** .
   Die Anzeige 'Computerverwaltung' wird geöffnet.
4. Erweitern Sie in der Anzeige 'Computerverwaltung' den Eintrag **Lokale Benutzer und Gruppen** .
5. Wählen Sie **Gruppen** aus.
6. Klicken Sie doppelt auf eine Gruppe. Die Anzeige mit den Gruppeneigenschaften wird angezeigt.
   Die Anzeige mit den Gruppeneigenschaften wird angezeigt.

Ergebnisse

Die Gruppenmitglieder werden angezeigt.

Windows *Benutzer unter Windows aus einer Gruppe entfernen*

Sie können einen Benutzer aus einer Gruppe entfernen, indem Sie die Steuerkonsole verwenden.

Vorgehensweise

1. Steuerkonsole öffnen

2. Klicken Sie doppelt auf **Verwaltung** .
   Die Anzeige mit den Verwaltungstools wird geöffnet.
3. Klicken Sie doppelt auf **Computerverwaltung** .
   Die Anzeige 'Computerverwaltung' wird geöffnet.
4. Erweitern Sie in der Anzeige 'Computerverwaltung' den Eintrag **Lokale Benutzer und Gruppen** .
5. Wählen Sie **Benutzer** aus.
6. Klicken Sie doppelt auf den Benutzer, der zu einer Gruppe hinzugefügt werden soll.
   Die Anzeige mit den Benutzereigenschaften wird angezeigt.
7. Wählen Sie die Registerkarte **Mitglied von** aus.
8. Wählen Sie die Gruppe aus, aus der Sie den Benutzer entfernen möchten, und klicken Sie dann auf **Entfernen** .
9. Klicken Sie auf **OK**.
   Die Anzeige 'Computerverwaltung' wird angezeigt.

Ergebnisse
Sie haben nun den Benutzer aus der Gruppe entfernt.

Windows Besondere Hinweise zur Sicherheit unter Windows

Einige Sicherheitsfunktionen verhalten sich in verschiedenen Versionen von Windows unterschiedlich.

Die IBM MQ-Sicherheit basiert auf Aufrufen an die Betriebssystem-API, in denen Informationen zu Benutzerberechtigungen und Gruppenzugehörigkeit angefordert werden. Einige Funktionen verhalten sich auf den Windows-Systemen nicht identisch. In dieser Themensammlung wird beschrieben, wie sich diese Unterschiede auf die IBM MQ-Sicherheit auswirken können, wenn Sie IBM MQ in einer Windows-Umgebung ausführen.

Windows *Lokale Konten und Domänenbenutzerkonten für den IBM MQ Windows-Service*
Bei der Ausführung von IBM MQ muss geprüft und sichergestellt werden, dass nur berechtigte Benutzer auf Warteschlangenmanager oder Warteschlangen zugreifen können. Dazu ist ein bestimmtes Benutzerkonto erforderlich, mit dem IBM MQ Informationen zu jedem Benutzer abfragen kann, der einen solchen Zugriff versucht.

- „Spezielle Benutzerkonten mit dem Prepare IBM MQ Wizarden konfigurieren” auf Seite 160
- „IBM MQ mit Active Directory verwenden” auf Seite 161
- „Erforderliche Benutzerberechtigungen für einen IBM MQ Windows-Service” auf Seite 161

Spezielle Benutzerkonten mit dem Prepare IBM MQ Wizarden konfigurieren

Der Prepare IBM MQ Wizard erstellt ein spezielles Benutzerkonto, damit der Windows-Service gemeinsam von Prozessen genutzt werden kann, die ihn verwenden müssen (siehe IBM MQ mit dem Prepare IBM MQ Wizard konfigurieren).

Ein Windows-Service wird von Clientprozessen für eine IBM MQ-Installation gemeinsam genutzt. Für jede Installation wird ein Service erstellt. Jeder Service hat den Namen `MQ_`*`InstallationName`* und den Anzeigenamen `IBM MQ(`*`InstallationName`*`)`.

Da jeder Service von nicht interaktiven und interaktiven Anmeldesitzungen gemeinsam genutzt werden muss, müssen Sie jeden Service unter einem speziellen Benutzeraccount starten. Sie können ein spezielles Benutzerkonto für alle Services verwenden oder verschiedene spezielle Benutzerkonten erstellen. Jedes spezielle Benutzerkonto muss über die Benutzerberechtigung `Anmelden als Dienst` verfügen. Weitere Informationen finden Sie unter Tabelle 14 auf Seite 161. Wenn die Benutzer-ID nicht über die Berechtigung zur Ausführung des Service verfügt, kann der Service nicht gestartet werden und im Windows-Systemereignisprotokoll wird ein Fehler gemeldet. Typischerweise haben Sie den Prepare IBM MQ

Wizarden ausgeführt und die Benutzer-ID ordnungsgemäß festgelegt. Wenn Sie die Benutzer-ID jedoch manuell konfiguriert haben, ist es möglich, dass Sie ein Problem haben, das Sie beheben müssen.

Wenn Sie IBM MQ installieren und den Prepare IBM MQ Wizarden das erste Mal ausführen, wird ein lokales Benutzerkonto für den Service mit der Bezeichnung MUSR_MQADMIN erstellt, das die erforderlichen Einstellungen und Berechtigungen, einschließlich `Anmelden als Dienst`, enthält.

In nachfolgenden Installationen erstellt der Prepare IBM MQ Wizard ein Benutzerkonto mit der Bezeichnung MUSR_MQADMIN*x*, wobei *x* für die nächste verfügbare Zahl steht und eine nicht vorhandene Benutzer-ID darstellt. Das Kennwort für MUSR_MQADMIN*x* wird beim Erstellen des Kontos zufällig generiert und zum Konfigurieren der Anmeldeumgebung für den Service verwendet. Das generierte Kennwort läuft nicht ab.

Dieses IBM MQ-Konto wird nicht von Kontorichtlinien beeinträchtigt, die im System eingerichtet sind und durch die Kennwörter für das Konto nach einem bestimmten Zeitraum geändert werden müssen.

Das Kennwort ist außerhalb dieser einmaligen Verarbeitung nicht bekannt und wird vom Windows-Betriebssystem in einem sicheren Teil der Registry gespeichert.

IBM MQ mit Active Directory verwenden

In einigen Netzkonfigurationen, in denen Benutzerkonten auf Domänencontrollern definiert sind, die den Active Directory-Verzeichnisservice verwenden, ist das lokale Benutzerkonto, unter dem IBM MQ ausgeführt wird, möglicherweise nicht zur Abfrage der Gruppenzugehörigkeit anderer Domänenbenutzerkonten berechtigt. Dies wird bei der Installation von IBM MQ vom Prepare IBM MQ Wizarden ermittelt, indem er Test durchführt und dem Benutzer Fragen zur Netzkonfiguration stellt.

Wenn das lokale Benutzerkonto, unter dem IBM MQ ausgeführt wird, nicht über die erforderliche Berechtigung verfügt, fordert der Prepare IBM MQ Wizard den Benutzer auf, die Kontodetails eines Domänenbenutzerkontos mit bestimmten Benutzerberechtigungen einzugeben. Informationen zum Erstellen und Einrichten eines Windows-Domänenkontos finden Sie unter Windows-Domänenkonten für IBM MQ erstellen und einrichten. Informationen zu den Benutzerberechtigungen, die für das Domänenbenutzerkonto erforderlich sind, finden Sie im Abschnitt Tabelle 14 auf Seite 161.

Nachdem der Benutzer die gültigen Kontodetails für das Domänenbenutzerkonto in den Prepare IBM MQ Wizarden eingegeben hat, konfiguriert der Assistent einen IBM MQ Windows-Service, der unter dem neuen Konto ausgeführt werden soll. Die Kontodetails werden im sicheren Teil der Registry festgehalten und können nicht von den Benutzern gelesen werden.

Wenn der Service ausgeführt wird, wird ein IBM MQ Windows-Service gestartet und bleibt so lange aktiv, wie der Service ausgeführt wird. Ein IBM MQ-Administrator, der sich nach dem Start des Windows-Service am Server anmeldet, kann die Warteschlangenmanager auf dem Server mit dem IBM MQ Explorer verwalten. Dadurch wird eine Verbindung zwischen dem IBM MQ Explorer und dem vorhandenen Windows-Serviceprozess hergestellt. Diese beiden Aktionen benötigen unterschiedliche Berechtigungsstufen, bevor sie funktionieren können:

- Für den Startprozess ist eine Startberechtigung erforderlich.
- Der IBM MQ-Administrator benötigt eine Zugriffsberechtigung.

Erforderliche Benutzerberechtigungen für einen IBM MQ Windows-Service

In der folgenden Tabelle werden die Benutzerberechtigungen angezeigt, die für die lokalen Konten und die Domänenbenutzerkonten erforderlich sind, unter denen der Windows-Service für eine IBM MQ-Installation ausgeführt wird.

Tabelle 14. Erforderliche Benutzerberechtigungen für einen IBM MQ Windows-Service

Erlaubnis	**Beschreibung**
Als Stapeljob anmelden	Aktiviert einen IBM MQ Windows-Service, der unter diesen Benutzerkonto ausgeführt werden soll.

Tabelle 14. Erforderliche Benutzerberechtigungen für einen IBM MQ Windows-Service (Forts.)	
Erlaubnis	**Beschreibung**
Als Dienst anmelden	Ermöglicht Benutzern, den IBM MQ Windows-Service für die Anmeldung unter Verwendung des konfigurierten Kontos festzulegen.
System herunterfahren	Ermöglicht dem IBM MQ Windows-Service bei entsprechender Konfiguration, den Server erneut zu starten, wenn ein Service nicht wiederhergestellt werden kann.
Kontingente erhöhen	Erforderlich für den Aufruf des Betriebssystems `CreateProcessAsUser` .
Einsetzen als Teil des Betriebssystems	Erforderlich für den Aufruf des Betriebssystems `LogonUser` .
Durchgangsprüfung umgehen	Erforderlich für den Aufruf des Betriebssystems `LogonUser` .
Ersetzen Sie ein Token auf Prozessebene.	Erforderlich für den Aufruf des Betriebssystems `LogonUser` .

Anmerkung: In Umgebungen, in der ASP-und IIS-Anwendungen ausgeführt werden, sind möglicherweise Debugprogrammberechtigungen erforderlich.

Für Ihr Domänenbenutzerkonto müssen diese Windows-Benutzerberechtigungen als effektive Benutzerberechtigungen festgelegt sein, die in der Anwendung "Lokale Sicherheitsrichtlinie" aufgeführt sind. Ist dies nicht der Fall, setzen Sie sie entweder lokal auf dem Server mit der Anwendung "Lokale Sicherheitsrichtlinie" oder in der Domäne "Domäne Security Application" (Domänensicherheitsanwendung).

Windows *Sicherheitsberechtigungen für den Windows-Server*

Bei der Installation von IBM MQ auf einem Windows-Server gibt es Unterschiede im Verhalten, je nachdem, ob ein lokaler Benutzer oder ein Domänenbenutzer die Installation ausführt.

Wenn ein *lokaler* Benutzer IBM MQinstalliert, erkennt Prepare IBM MQ Wizard , dass der für den IBM MQ Windows -Service erstellte lokale Benutzer die Gruppenzugehörigkeitsinformationen des installierenden Benutzers abrufen kann. Der Prepare IBM MQ Wizard befragt den Benutzer zur Netzkonfiguration, um zu ermitteln, ob auf Domänencontroller, die unter Windows 2000 oder höher ausgeführt werden, noch weitere Benutzerkonten definiert sind. Ist dies der Fall, muss der IBM MQ Windows-Service unter einem Domänenbenutzerkonto mit bestimmten Einstellungen und Berechtigungen ausgeführt werden. Der Prepare IBM MQ Wizard fordert den Benutzer zur Eingabe der Kontodetails für diesen Benutzer auf, wie unter IBM MQ mit dem Prepare IBM MQ Wizard konfigurieren beschrieben.

Wenn ein *Domänenbenutzer* IBM MQinstalliert, erkennt Prepare IBM MQ Wizard , dass der lokale Benutzer, der für den IBM MQ Windows -Service erstellt wurde, die Gruppenzugehörigkeitsinformationen des installierenden Benutzers nicht abrufen kann. In diesem Fall fordert der Prepare IBM MQ Wizard den Benutzer immer zur Eingabe der Kontodetails für das Domänenbenutzerkonto auf, das vom IBM MQ Windows-Service verwendet werden soll.

Wenn der IBM MQ Windows-Service ein Domänenbenutzerkonto verwenden muss, kann IBM MQ erst dann ordnungsgemäß ausgeführt werden, der Prepare IBM MQ Wizard dieses Konto konfiguriert hat. Der Prepare IBM MQ Wizard erlaubt dem Benutzer erst dann, mit anderen Tasks fortzufahren, wenn der Windows-Service mit einem geeigneten Konto konfiguriert wurde.

Weitere Informationen finden Sie unter Erstellen und Einrichten von Domänenkonten für IBM MQ.

Windows *Benutzernamen ändern, der dem IBM MQ-Service zugeordnet ist*

Die können den Benutzernamen ändern, der dem IBM MQ-Service zugeordnet ist, indem Sie ein neues Konto erstellen und die entsprechenden Einzelheiten mithilfe des Prepare IBM MQ Wizarden eingeben.

Informationen zu diesem Vorgang

Bei der Installation von IBM MQ und der ersten Ausführung des Prepare IBM MQ Wizarden wird ein lokales Benutzerkonto für den Service mit der Bezeichnung MUSR_MQADMIN erstellt. In nachfolgenden Installationen erstellt der Prepare IBM MQ Wizard ein Benutzerkonto mit der Bezeichnung MUSR_MQADMIN*x*, wobei *x* für die nächste verfügbare Zahl steht und eine nicht vorhandene Benutzer-ID darstellt.

Möglicherweise müssen Sie den Benutzernamen, der dem IBM MQ-Service zugeordnet ist, von MUSR_MQADMIN oder MUSR_MQADMIN*x* in eine andere Bezeichnung ändern. Dies kann beispielsweise erforderlich sein, wenn Ihr Warteschlangenmanager Db2 zugeordnet ist, für das keine Benutzernamen mit mehr als 8 Zeichen zulässig sind.

Vorgehensweise

1. Erstellen Sie ein neues Benutzerkonto (z. B. **NEW_NAME**).
2. Im Prepare IBM MQ Wizarden können Sie die Einzelheiten des neuen Benutzerkontos eingeben.

Zugehörige Tasks

IBM MQ mit dem Prepare IBM MQ Wizard konfigurieren

Windows *Kennwort des lokalen Benutzerkontos für den IBM MQ Windows-Service ändern*

Sie können das Kennwort des lokalen Benutzerkontos für den IBM MQ Windows-Service in der Anzeige 'Computerverwaltung' ändern.

Informationen zu diesem Vorgang

Um das Kennwort des lokalen Benutzerkontos für den IBM MQ Windows-Service zu ändern, führen Sie die folgenden Schritte aus:

Vorgehensweise

1. Geben Sie den Benutzer an, unter dem der Service ausgeführt wird.
2. Stoppen Sie den IBM MQ-Service in der Anzeige 'Computerverwaltung'.
3. Ändern Sie das erforderliche Kennwort auf die gleiche Weise wie das Kennwort einer Person.
4. Rufen Sie die Eigenschaften für den IBM MQ-Service in der Anzeige 'Computerverwaltung' auf.
5. Wählen Sie die Seite **Anmelden** aus.
6. Bestätigen Sie, dass der angegebene Accountname mit dem Benutzer übereinstimmt, für den das Kennwort geändert wurde.
7. Geben Sie das Kennwort in die Felder **Kennwort** und **Kennwort bestätigen** ein, und klicken Sie auf **OK** .

Windows *Kennwort für einen IBM MQ Windows-Service für die Installation unter einem Domänenbenutzerkonto ändern*

Alternativ zur Verwendung des Prepare IBM MQ Wizarden zur Eingabe von Kontodetails für das Domänenbenutzerkonto können Sie in der Anzeige 'Computerverwaltung' die Einzelheiten für die **Anmeldung** für den entsprechenden IBM MQ-Service für die Installation ändern.

Informationen zu diesem Vorgang

Wenn der IBM MQ Windows-Service für eine Installation unter einem Domänenbenutzerkonto ausgeführt wird, können Sie das Kennwort für das Konto folgendermaßen ändern:

Vorgehensweise

1. Ändern Sie das Kennwort für das Domänenkonto auf dem Domänencontroller. Möglicherweise müssen Sie Ihren Domänenadministrator bitten, dies für Sie zu tun.

2. Führen Sie die folgenden Schritte aus, um die Seite **Log On** (Anmeldung) für den IBM MQ-Service zu ändern.
   a) Geben Sie den Benutzer an, unter dem der Service ausgeführt wird.
   b) Stoppen Sie den IBM MQ-Service in der Anzeige 'Computerverwaltung'.
   c) Ändern Sie das erforderliche Kennwort auf die gleiche Weise wie das Kennwort einer Person.
   d) Rufen Sie die Eigenschaften für den IBM MQ-Service in der Anzeige 'Computerverwaltung' auf.
   e) Wählen Sie die Seite **Anmelden** aus.
   f) Bestätigen Sie, dass der angegebene Accountname mit dem Benutzer übereinstimmt, für den das Kennwort geändert wurde.
   g) Geben Sie das Kennwort in die Felder **Kennwort** und **Kennwort bestätigen** ein, und klicken Sie auf **OK** .

   Das Benutzerkonto, unter dem der IBM MQ Windows-Service ausgeführt wird, führt alle MQSC-Befehle aus, die von Benutzerschnittstellenanwendungen ausgegeben werden oder die beim Starten und Beenden des Systems oder bei der Servicewiederherstellung automatisch ausgeführt werden. Dieses Benutzerkonto muss deshalb über Administratorberechtigungen für IBM MQ verfügen. Es wird standardmäßig der lokalen Gruppe 'mqm' auf dem Server hinzugefügt. Wenn diese Mitgliedschaft entfernt wird, funktioniert der IBM MQ Windows-Service nicht. Weitere Informationen zu Benutzerberechtigungen finden Sie unter „Erforderliche Benutzerberechtigungen für einen IBM MQ Windows-Service” auf Seite 161.

   Tritt ein Sicherheitsproblem mit dem Benutzerkonto auf, unter dem der IBM MQ Windows-Service ausgeführt wird, werden Fehlernachrichten und Beschreibungen im Systemereignisprotokoll angezeigt.

Zugehörige Tasks
IBM MQ mit dem Prepare IBM MQ Wizard konfigurieren

Windows *Hinweise zum Hochstufen von Windows-Servern zu Domänencontrollern*

Wenn Sie einen Windows-Server zu einem Domänencontroller hochstufen, sollten Sie entscheiden, ob die Sicherheitseinstellung für die Benutzer- und Gruppenberechtigungen noch geeignet ist. Wenn der Status einer Windows-Maschine zwischen einem Server und einem Domänencontroller geändert wird, müssen Sie beachten, dass sich dies auf die Operation von IBM MQ auswirken kann, da IBM MQ die lokal definierte Gruppe 'mqm' verwendet.

Sicherheitseinstellungen für Domänenbenutzer und Gruppenberechtigungen

IBM MQ basiert darauf, dass die erforderliche Sicherheitsrichtlinie von den Informationen zur Gruppenzugehörigkeit implementiert werden, was bedeutet, dass die Benutzer-ID, mit der IBM MQ-Operationen ausgeführt werden, die Gruppenzugehörigkeit anderer Benutzer ermitteln kann.

Beim Hochstufen eines Windows-Servers zu einem Domänencontroller wird Ihnen eine Option für die Sicherheitseinstellungen angezeigt, die sich auf Benutzer- und Gruppenberechtigungen beziehen. Mit dieser Option wird gesteuert, ob beliebige Benutzer Gruppenzugehörigkeiten aus dem aktiven Verzeichnis abrufen können. Wenn ein Domänencontroller so konfiguriert ist, dass lokale Konten zur Abfrage der Gruppenzugehörigkeit von Domänenbenutzerkonten berechtigt sind, kann die von IBM MQ während des Installationsprozesses erstellte standardmäßige Benutzer-ID bei Bedarf Gruppenzugehörigkeiten für andere Benutzer abrufen. Wenn ein Domänencontroller allerdings so konfiguriert ist, dass lokale Konten nicht zur Abfrage der Gruppenzugehörigkeit von Domänenbenutzerkonten berechtigt sind, kann IBM MQ nicht abschließend überprüfen, ob Benutzer, die in der Domäne definiert sind, für den Zugriff auf Warteschlangenmanager und Warteschlangen berechtigt sind, und der Zugriff schlägt fehl. Wenn Sie Windows auf einem Domänencontroller verwenden, der auf diese Weise konfiguriert ist, muss ein spezielles Domänenbenutzerkonto mit den erforderlichen Berechtigungen verwendet werden.

In diesem Fall müssen Sie Folgendes wissen:

- Wie verhalten sich Sicherheitsberechtigungen für Ihre Version von Windows?

- Wie erhalten Mitglieder der Domänengruppe 'mqm' die Berechtigung zum Lesen der Gruppenzugehörigkeit?
- Wie wird ein IBM MQ Windows-Service für die Ausführung unter einem Domänenbenutzer konfiguriert?

Weitere Informationen finden Sie unter Benutzerkonten für IBM MQ konfigurieren.

IBM MQ-Zugriff auf die lokale mqm-Gruppe

Wenn Windows-Server zu Domänencontroller hoch- oder herabstuft werden, verliert IBM MQ den Zugriff auf die lokale mqm-Gruppe.

Wenn ein Server als Domänencontroller hochgestuft wird, ändert sich der Geltungsbereich von der lokalen in die lokale Domäne. Wenn die Maschine auf den Server herabgestuft wird, werden alle lokalen Gruppen-Gruppen entfernt. Dies bedeutet, dass das Ändern einer Maschine vom Server zum Domänencontroller und zurück zum Server den Zugriff auf eine lokale mqm-Gruppe verliert. Das Symptom ist ein Fehler, der den Mangel an einer lokalen mqm-Gruppe angibt, z. B.:

```
>crtmqm qm0
AMQ8066:Local mqm group not found.
```

Um dieses Problem zu beheben, erstellen Sie die lokale Gruppe 'mqm' mit den Standardverwaltungstools von Windows erneut. Da alle Informationen zur Gruppenzugehörigkeit verloren gegangen sind, müssen Sie privilegierte IBM MQ-Benutzer in der neu erstellten lokalen Gruppe 'mqm' erneut wiederherstellen. Wenn es sich bei der Maschine um ein Domänenmitglied handelt, müssen Sie außerdem die mqm-Gruppe für die Domäne der lokalen mqm-Gruppe hinzufügen, um privilegierten IBM MQ-Benutzer-IDs der Domäne die erforderliche Berechtigungsebene zu erteilen.

Windows *Einschränkungen für verschachtelte Gruppen in Windows*

Es gibt Einschränkungen bei der Verwendung von verschachtelten Gruppen. Diese begründen sich teilweise auf Einschränkungen auf Domänenfunktionsebene, teilweise auf Einschränkungen seitens IBM MQ.

Active Directory kann verschiedene Gruppentypen in einem Domänenkontext unterstützen, abhängig von der Domänenfunktionsebene. Windows 2003-Domänen befinden sich standardmäßig auf der Funktionsebene " Windows 2000 gemischt". (Windows Server 2008 und Windows Server 2012 folgen dem Windows 2003-Domänenmodell.) Die funktionale Ebene der Domäne bestimmt die unterstützten Gruppentypen und die Verschachtetungsebene, die bei der Konfiguration von Benutzer-IDs in einer Domänenumgebung zulässig ist. Ausführliche Informationen zu den Kriterien für den Gruppenumfang und das Einschlusskriterium finden Sie in der Active Directory-Dokumentation

Neben den Voraussetzungen für Active Directory gelten für die von IBM MQ verwendeten IDs zusätzliche Einschränkungen. Die von IBM MQ verwendeten Netz-APIs unterstützen nicht alle Konfigurationen, die auf Domänenfunktionsebene unterstützt werden. Daher kann IBM MQ keine Gruppenzugehörigkeiten von Domänen-IDs abfragen, die sich in einer lokalen Domänengruppe befinden, die wiederum in einer lokalen Gruppe verschachtelt ist. Darüber hinaus wird die Mehrfachverschachtnierung von globalen und universellen Gruppen nicht unterstützt. Es werden jedoch sofort verschachtelte globale oder universelle Gruppen unterstützt.

Windows *Benutzern die ferne Verwendung von IBM MQ ermöglichen*

Wenn Sie beim Herstellen einer fernen Verbindung zu IBM MQ Warteschlangenmanager erstellen und starten müssen, müssen Sie über den Benutzerzugriff `Create global objects` (Globale Objekte erstellen) verfügen.

Informationen zu diesem Vorgang

Anmerkung: Administratoren verfügen standardmäßig über den Benutzerzugriff `Globale Objekte erstellen`. Als Administrator können Sie also ohne Änderung Ihrer Benutzerberechtigungen über Remotezugriff Warteschlangenmanager erstellen und starten.

Wenn Sie eine Verbindung zu einem Windows-System mithilfe von Terminal Services oder einer Remote Desktop-Verbindung herstellen und es beim Erstellen, Starten oder Löschen eines Warteschlangenmanagers Probleme gibt, kann dies am fehlenden Benutzerzugriff `Create global objects` liegen.

Durch den Benutzerzugriff `Create global objects` werden die Benutzer begrenzt, die berechtigt sind, Objekte im globalen Namensbereich zu erstellen. Eine Anwendung kann nur dann ein globales Objekt erstellen, wenn sie im globalen Namensbereich ausgeführt wird oder wenn dem Benutzer, der die Anwendung ausführt, der Benutzerzugriff `Create global objects` zugeordnet ist.

Wenn Sie über Teminal Services oder eine Remote Desktop-Verbindung über Fernzugriff mit einem Windows-System verbunden sind, werden Anwendungen in ihrem eigenen lokalen Namensbereich ausgeführt. Wenn Sie versuchen, einen Warteschlangenmanager mit IBM MQ Explorer oder dem Befehl **crtmqm** oder **dltmqm** zu erstellen bzw. zu löschen oder einen Warteschlangenmanager mit dem Befehl **strmqm** zu starten, führt das zu einem Berechtigungsfehler. Es wird eine IBM MQ-FDC-Datei mit der Ereignis-ID XY132002 erstellt.

Ein Warteschlangenmanager kann problemlos über IBM MQ Explorer oder mit dem Befehl **amqmdain qmgr start** gestartet werden, da der Warteschlangenmanager auf diese Weise nicht direkt gestartet wird. Die Befehle senden die Anforderung zum Starten des Warteschlangenmanagers an einen gesonderten Prozess, der im globalen Namensbereich ausgeführt wird.

Wenn die verschiedenen Methoden zur Verwaltung von IBM MQ bei der Verwendung von Terminal Services nicht funktionieren, legen Sie die Benutzerberechtigung `Create global objects` fest.

Vorgehensweise

1. Öffnen Sie die Anzeige Verwaltungstools:

   Windows Server 2008 und Windows Server 2012
   Öffnen Sie dieses Tool über die Menüfolge **Systemsteuerung** > **System und Wartung** > **Verwaltung**.

   Windows 8.1
   Öffnen Sie diese Anzeige über die Menüfolge **Verwaltung** > **Systemsteuerung**
2. Klicken Sie doppelt auf **Lokale Sicherheitsrichtlinie** .
3. Erweitern Sie **Lokale Richtlinien** .
4. Klicken Sie auf **Zuweisen von Benutzerrechten** .
5. Fügen Sie den neuen Benutzer oder die neue Gruppe zur Richtlinie `Create global objects` hinzu.

Windows *SSPI-Kanalexitprogramm unter Windows*

IBM MQ for Windows stellt ein Sicherheitsexitprogramm bereit, das auf Nachrichten- und MQI-Kanälen verwendet werden kann. Der Exit wird als Quellen-und Objektcode bereitgestellt und stellt eine Einweg- und eine Zwei-Wege-Authentifizierung zur Verfügung.

Der Sicherheitsexit verwendet die Security Support Provider Interface (SSPI), mit der die integrierten Sicherheitsfunktionen von Windows-Plattformen bereitgestellt werden.

Der Sicherheitsexit stellt die folgenden Identifizierungs-und Authentifizierungsservices bereit:

Einweg-Authentifizierung
Hierbei wird die Unterstützung für die Authentifizierung durch den Windows NT LAN Manager (NTLM) verwendet. NTLM ermöglicht es Servern, ihre Clients zu authentifizieren. Es erlaubt einem Client nicht, einen Server zu authentifizieren, oder einen Server, um einen anderen zu authentifizieren. NTLM wurde für eine Netzumgebung konzipiert, in der die Server als echt gelten. NTLM wird auf allen Windows -Plattformen unterstützt, die von IBM WebSphere MQ 7.0 unterstützt werden.

Dieser Service wird in der Regel in einem MQI-Kanal verwendet, um die Authentifizierung einer IBM MQ MQI client-Anwendung durch einen Serverwarteschlangenmanager zu ermöglichen. Eine Clientanwendung wird durch die Benutzer-ID identifiziert, die dem Prozess zugeordnet ist, der ausgeführt wird.

Um die Authentifizierung durchzuführen, fordert der Sicherheitsexit auf der Clientseite eines Kanals ein Authentifizierungstoken von NTLM an und sendet das Token in einer Sicherheitsnachricht an seinen Partner am anderen Ende des Kanals. Der Sicherheitsexit der Partnersicherheit übergibt das Token an NTLM, das prüft, ob das Token authentisch ist. Wenn der Sicherheitsexit der Partnerverbindung nicht mit der Authentizität des Tokens zufrieden ist, weist er den MCA an, den Kanal zu schließen.

Zwei-Wege-Authentifizierung oder gegenseitige Authentifizierung

Dies verwendet Kerberos-Authentifizierungsservices. Das Kerberos-Protokoll nimmt nicht an, dass die Server in einer Netzumgebung echt sind. Server können Clients und andere Server authentifizieren, und Clients können Server authentifizieren. Kerberos wird auf allen Windows-Plattformen unterstützt, die von IBM WebSphere MQ 7.0 unterstützt werden.

Dieser Service kann sowohl für Nachrichten-als auch für MQI-Kanäle verwendet werden. In einem Nachrichtenkanal wird die gegenseitige Authentifizierung der beiden WS-Manager bereitgestellt. In einem MQI-Kanal können der Serverwarteschlangenmanager und die IBM MQ MQI client-Anwendung sich dadurch gegenseitig authentifizieren. Ein Warteschlangenmanager wird durch seinen Namen identifiziert, der durch die Zeichenfolge `ibmMQSeries/` vorangestellt ist. Eine Clientanwendung wird durch die Benutzer-ID identifiziert, die dem Prozess zugeordnet ist, der ausgeführt wird.

Um die gegenseitige Authentifizierung durchzuführen, fordert der einleitende Sicherheitsexit ein Authentifizierungstoken vom Kerberos-Sicherheitsserver an und sendet das Token in einer Sicherheitsnachricht an seinen Partner. Der Sicherheitsexit der Partnersicherheit übergibt das Token an den Kerberos-Server, der authentisch überprüft. Der Kerberos-Sicherheitsserver generiert ein zweites Token, das der Partner in einer Sicherheitsnachricht an den einleitenden Sicherheitsexit sendet. Der einleitende Sicherheitsexit fordert den Kerberos-Server dann auf, zu überprüfen, ob das zweite Token authentisch ist. Wenn der Sicherheitsexit bei diesem Austausch nicht mit der Authentizität des von der anderen gesendeten Tokens zufrieden ist, weist er den MCA an, den Kanal zu schließen.

Der Sicherheitsexit wird sowohl im Quellen-als auch im Objektformat angegeben. Sie können den Quellcode als Ausgangspunkt zum Schreiben eigener Kanalexitprogramme verwenden oder Sie können das Objektmodul wie angegeben verwenden. Das Objektmodul hat zwei Eingangspunkte, eine für die eine Art der Authentifizierung, die die NTLM-Authentifizierungsunterstützung verwendet, und die andere für die Zweiwege-Authentifizierung unter Verwendung von Kerberos-Authentifizierungsservices.

Weitere Informationen zur Funktionsweise des SSPI-Kanalexitprogramms und Anweisungen zur Implementierung finden Sie unter SSPI-Sicherheitsexit auf Windows-Systemen verwenden.

Windows *Sicherheitsschablonendateien unter Windows anwenden*

Die Anwendung einer Schablone kann sich auf die Sicherheitseinstellungen auswirken, die für IBM MQ-Dateien und -Verzeichnisse angewendet werden. Wenn Sie die Schablone 'Highly Secure) (Sehr sicher) verwenden, wenden Sie diese vor der Installation von IBM MQ an.

Windows unterstützt textbasierte Sicherheitsschablonendateien, mit denen Sie einheitliche Sicherheitseinstellungen auf einem oder mehreren Computern über das MMC-Snap-in 'Security Configuration and Analysis' (Sicherheitskonfiguration und Analyse) anwenden können. Windows bietet verschiedene Schablonen mit einem breiten Spektrum an Sicherheitseinstellungen an, die bestimmte Sicherheitsstufen bereitstellen. Zu diesen Schablonen gehören Compatible, Secure und Highly Secure.

Wenn Sie eine dieser Schablonen anwenden, kann sich dies auf die für IBM MQ-Dateien und -Verzeichnisse geltenden Sicherheitseinstellungen auswirken. Wenn Sie die Schablone 'Highly Secure' verwenden möchten, konfigurieren Sie Ihre Maschine vor der Installation von IBM MQ.

Wenn Sie die Schablone 'Highly Secure' auf einer Maschine anwenden möchten, auf der IBM MQ bereits installiert ist, werden alle Berechtigungen entfernt, die Sie in den IBM MQ-Dateien und -Verzeichnissen festgelegt haben. Da diese Berechtigungen entfernt werden, verlieren Sie *Administrator* , *mqm* und, falls zutreffend, den Gruppenzugriff *Jeder* aus den Fehlerverzeichnissen.

Windows *Zusatzberechtigung für Windows-Anwendungen konfigurieren, die eine Verbindung zu IBM MQ herstellen*

Das Konto, unter dem IBM MQ-Prozesse ausgeführt werden, benötigt möglicherweise eine zusätzliche Berechtigung, damit der Zugriff SYNCHRONIZE auf Anwendungsprozesse erteilt werden kann.

Informationen zu diesem Vorgang

Es können Probleme auftreten, wenn Windows-Anwendungen (z. B. ASP-Seiten), die eine Verbindung zu IBM MQ herstellen, so konfiguriert sind, dass sie auf einer höheren Sicherheitsebene als üblich ausgeführt werden.

Für IBM MQ ist der Zugriff SYNCHRONIZE auf Anwendungsprozesse erforderlich, damit bestimmte Aktionen koordiniert werden können. Beim ersten Versuch einer Serveranwendung, eine Verbindung zu einem Warteschlangenmanager herzustellen, wird in IBM MQ der Prozess geändert, in dem IBM MQ-Administratoren in die Berechtigung SYNCHRONIZE erteilt wird. Der Account, unter dem IBM MQ-Prozesse ausgeführt werden, benötigt jedoch möglicherweise zusätzliche Berechtigungen, bevor der angeforderte Zugriff erteilt werden kann.

Führen Sie die folgenden Schritte aus, um die Zusatzberechtigung für die Benutzer-ID zu konfigurieren, unter der IBM MQ-Prozesse ausgeführt werden:

Vorgehensweise

1. Starten Sie das Tool 'Local Security Policy' (Lokale Sicherheitsrichtlinie), klicken Sie auf **Security Settings**->**Local Policies**->**User Right Assignments** (Sicherheitseinstellungen > Lokale Richtlinien > Zuordnungen zur Benutzerberechtigung) und klicken Sie anschließend auf **Debug Programs** (Debugprogramme).
2. Klicken Sie doppelt auf **Debug Programs** und fügen Sie der Liste Ihre IBM MQ-Benutzer-ID hinzu.

   Wenn sich das System in einer Windows-Domäne befindet und die effektive Richtlinie noch nicht eingerichtet ist, obwohl die lokale Richtlinieneinstellung vorgenommen wurde, muss die Benutzer-ID auf die gleiche Weise mit dem Tool 'Sicherheitsrichtlinie der Domäne' auf Domänenebene autorisiert werden.

IBM i Sicherheit unter IBM i einrichten

Die Sicherheit unter IBM i wird mithilfe des Objektberechtigungsmanagers (OAM) für IBM MQ und der IBM i-Sicherheit auf Objektebene implementiert.

Sicherheitsaspekte, die bei der Einrichtung der Zugriffsberechtigung für IBM MQ-Objekte berücksichtigt werden müssen.

Sie müssen die folgenden Punkte berücksichtigen, wenn Sie die Berechtigungen für die Benutzer in Ihrem Unternehmen einrichten:

1. Erteilen und entziehen Sie Berechtigungen für die IBM MQ for IBM i -Befehle mit den Befehlen IBM i GRTOBJAUT und RVKOBJAUT .

   In der QMQM-Bibliothek sind bestimmte Nicht-Befehlsobjekte (* cmd) so definiert, dass sie die Berechtigung ***PUBLIC** für ***USE** haben. Ändern Sie die Berechtigungen dieser Objekte nicht, oder verwenden Sie eine Berechtigungsliste, um die Berechtigung bereitzustellen. Falsche Berechtigungen können dazu führen, dass die Funktionen von IBM MQ nicht mehr richtig funktionieren.
2. Während der Installation von IBM MQ for IBM i werden die folgenden speziellen Benutzerprofile erstellt:

   QMQM
   Wird hauptsächlich für interne Produktfunktionen verwendet. Es kann jedoch verwendet werden, um vertrauenswürdige Anwendungen unter Verwendung von MQCNO_FASTPATH_BINDINGS auszuführen. Weitere Informationen finden Sie unter Verbindung zu einem Warteschlangenmanager mit dem MQCONNX-Aufruf herstellen .

QMQMADM
Wird als Gruppenprofil für Administratoren von IBM MQ verwendet. Das Gruppenprofil ermöglicht den Zugriff auf CL-Befehle und IBM MQ-Ressourcen.

Bei der Verwendung von SBMJOB zur Übergabe von Programmen, mit denen IBM MQ-Befehle aufgerufen werden, darf der Wert USER nicht ausdrücklich auf QMQMADM gesetzt werden. Stattdessen setzen Sie USER auf QMQM oder ein anderes Benutzerprofil, für das QMQMADM als Gruppe angegeben wurde.

3. Wenn Sie Kanalbefehle an ferne WS-Manager senden, müssen Sie sicherstellen, dass Ihr Benutzerprofil Mitglied der Gruppe QMQMADM auf dem Zielsystem ist. Eine Liste der PCF- und MQSC-Kanalbefehle finden Sie unter IBM MQ for IBM i-CL-Befehle.
4. Der Gruppensatz, der einem Benutzer zugeordnet ist, wird zwischengespeichert, wenn die Gruppenberechtigungen vom OAM berechnet werden.

   Alle Änderungen, die an den Gruppenzugehörigkeiten eines Benutzers vorgenommen werden, nachdem die Gruppengruppe in den Cache gestellt wurde, werden erst erkannt, wenn Sie den Warteschlangenmanager erneut starten oder RFRMQMAUT ausführen, um die Sicherheit zu aktualisieren .
5. Begrenzen Sie die Anzahl der Benutzer, die berechtigt sind, mit Befehlen zu arbeiten, die besonders empfindlich sind. Zu diesen Befehlen gehören:
   - Nachrichtenwarteschlangenmanager erstellen (`CRTMQM`)
   - Nachrichtenwarteschlangenmanager löschen (`DLTMQM`)
   - Nachrichtenwarteschlangenmanager starten (`STRMQM`)
   - Nachrichtenwarteschlangenmanager beenden (`ENDMQM`)
   - Befehlsserver starten (`STRMQMCSVR`)
   - Befehlsserver beenden (`ENDMQMCSVR`)
6. Kanaldefinitionen enthalten eine Spezifikation des Sicherheitsexitprogramms. Kanalerstellung und -änderung erfordert besondere Überlegungen. Ausführliche Informationen zu Sicherheitsexits finden Sie im Abschnitt „Übersicht über Sicherheitsexits” auf Seite 118.
7. Der Kanalexit und die Auslösermonitorprogramme können ersetzt werden. Die Sicherheit eines solchen Ersatzes liegt in der Verantwortung des Programmierers.

IBM i Objektberechtigungsmanager unter IBM i

Der Objektberechtigungsmanager (OAM) verwaltet die Berechtigungen von Benutzern, um IBM MQ-Objekte zu bearbeiten, einschließlich Warteschlangen und Prozessdefinitionen. Es stellt auch eine Befehlsschnittstelle zur Verfügung, über die Sie Zugriffsberechtigungen für ein Objekt für eine bestimmte Benutzergruppe erteilen oder entziehen können. Die Entscheidung für den Zugriff auf eine Ressource wird vom OAM getroffen, und der WS-Manager folgt dieser Entscheidung. Wenn der OAM keine Entscheidung treffen kann, verhindert der WS-Manager den Zugriff auf diese Ressource.

Über den OAM können Sie Folgendes steuern:

- Zugriff auf IBM MQ-Objekte über die MQI. Wenn ein Anwendungsprogramm versucht, auf ein Objekt zuzugreifen, prüft der OAM, ob das Benutzerprofil, das die Anforderung stellt, die Berechtigung für die angeforderte Operation hat.

  Dies bedeutet insbesondere, dass Warteschlangen und die Nachrichten in Warteschlangen vor unbefugtem Zugriff geschützt werden können.
- Berechtigung zum Verwenden von PCF-und MQSC-Befehlen.

Unterschiedliche Benutzergruppen können unterschiedliche Zugriffsberechtigungen für dasselbe Objekt haben. Für eine bestimmte Warteschlange kann eine Gruppe beispielsweise sowohl put-als auch get-Operationen ausführen. Eine andere Gruppe ist möglicherweise nur zum Durchsuchen der Warteschlange berechtigt (MQGET mit Suchoption). In ähnlicher Weise haben einige Gruppen möglicherweise die Be-

rechtigung zum Abrufen und zum Löschen von Berechtigungen für eine Warteschlange, aber es ist nicht zulässig, die Warteschlange zu ändern oder zu löschen.

IBM MQ for IBM i-Befehle und Operationen in IBM MQ for IBM i-Objekten ausführen

IBM i IBM MQ-Berechtigungen unter IBM i

Für den Zugriff auf IBM MQ-Objekte benötigen Sie die Berechtigung zur Ausgabe des Befehls und zum Zugriff auf das jeweilige Objekt. Administratoren können auf alle IBM MQ-Ressourcen zugreifen.

Der Zugriff auf IBM MQ-Objekte wird von den Berechtigungen für folgende Aufgaben gesteuert:

1. Ausgabe des IBM MQ-Befehls
2. Zugriff auf die IBM MQ-Objekte, auf die durch den Befehl verwiesen wird

Alle IBM MQ for IBM i-CL-Befehle werden mit dem Eigner QMQM geliefert, und das Verwaltungsprofil (QMQMADM) verfügt über die Berechtigung *USE, wobei der Zugriff *PUBLIC auf *EXCLUDE gesetzt ist.

Anmerkung: Das lizenzierte Programm zur Installation von IBM MQ for IBM i verwendet das Programm QSRDUPER, um Befehlsobjekte (*CMD) in QSYS zu duplizieren. In IBM i V5R4 und höher wurde das Programm QSRDUPER geändert, so dass als Standardverhalten nicht der ursprüngliche Befehl dupliziert, sondern ein Proxy-Befehl erstellt wird. Ein Proxy-Befehl leitet die Befehlsausführung an einen anderen Befehl um und weist ein Attribut von PRX auf. Wenn ein Proxy-Befehl mit demselben Namen wie der zu kopierende Befehl in der Bibliothek QSYS vorhanden ist, werden dem Befehl in der Produktbibliothek nicht die persönlichen Berechtigungen für den Proxy-Befehl erteilt. Es wird versucht, den Proxy-Befehl in QSYS anzufordern oder auszuführen und die Berechtigung des Zielbefehls in der Produktbibliothek zu überprüfen. Alle Änderungen der Berechtigung für *CMD-Objekte müssen daher in der Produktbibliothek (QMQM) vorgenommen werden, und die in QSYS müssen nicht geändert werden. Beispiel:

```
GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
```

Durch Änderungen an der Berechtigungsstruktur einiger CL-Befehle für das Produkt können diese Befehle öffentlich verwendet werden, wenn Sie über die erforderliche OAM-Berechtigung für die IBM MQ-Objekte verfügen, um diese Änderungen vornehmen zu können.

Als IBM MQ-Administrator unter IBM i müssen Sie ein Mitglied der *Gruppe QMQMADM* sein. Diese Gruppe verfügt über Eigenschaften, die den Eigenschaften der Gruppe 'mqm' auf AIX, Linux, and Windows-Systemen ähneln. Insbesondere wird die Gruppe QMQMADM bei der Installation von IBM MQ for IBM i erstellt und die Mitglieder der Gruppe QMQMADM haben Zugriff auf alle IBM MQ-Ressourcen im System. Sie haben auch Zugriff auf alle IBM MQ-Ressourcen, wenn Sie die Berechtigung *ALLOBJ haben.

Administratoren können CL-Befehle verwenden, um IBM MQ zu verwalten. Einer dieser Befehle ist GRTMQMAUT, der für die Erteilung von Berechtigungen für andere Benutzer verwendet wird. Ein anderer Befehl, STRMQMMQSC, ermöglicht es einem Administrator, MQSC-Befehle an einen lokalen WS-Manager auszugeben.

Zugehörige Konzepte

„Berechtigung für die Verwaltung von IBM MQ unter IBM i” auf Seite 95

IBM i *Zugriffsberechtigungen für IBM MQ-Objekte unter IBM i*

Zugriffsberechtigungen, die für die Ausführung von CL-Befehlen für IBM MQ erforderlich sind.

In IBM MQ for IBM i werden die CL-Befehle des Produkts in zwei Gruppen kategorisiert:

Gruppe 1

Benutzer müssen sich in der Benutzergruppe QMQMADM befinden oder über die Berechtigung *ALLOBJ verfügen, um diese Befehle verarbeiten zu können. Benutzer mit einer dieser Berechtigungen können alle Befehle in allen Kategorien verarbeiten, ohne dass eine zusätzliche Berechtigung erforderlich ist.

Anmerkung: Diese Berechtigungen überschreiben jede OAM-Berechtigung.

Diese Befehle können wie folgt gruppiert werden:

- Befehlsserverbefehle
  - ENDMQMCSVR, IBM MQ-Befehlsserver beenden
  - STRMQMCSVR, IBM MQ-Befehlsserver starten
- Befehl "Dead-Letter Queue Handler"
  - STRMQMDLQ, IBM MQ-Steuerroutine der Warteschlange für nicht zustellbare Nachrichten starten
- Listenerbefehl
  - ENDMQMLSR, IBM MQ-Listener beenden
  - STRMQMLSR, Nicht-Objekt-Listener starten
- Datenträgerwiederherstellungsbefehle
  - RCDMQMIMG, IBM MQ -Objektimage aufzeichnen
  - RCRMQMOBJ, IBM MQ-Objekt erneut erstellen
  - WRKMQMTRN, mit IBM MQ Q-Transaktionen arbeiten
- WS-Manager-Befehle
  - CRTMQM, Nachrichten-WS-Manager erstellen
  - DLTMQM, Nachrichten-WS-Manager löschen
  - ENDMQM, Nachrichten-WS-Manager beenden
  - STRMQM, Nachrichten-WS-Manager starten
- Sicherheitsbefehle
  - GRTMQMAUT, IBM MQ-Objektberechtigung erteilen
  - RVKMQMAUT, IBM MQ-Objektberechtigung widerrufen
- Trace-Befehl
  - TRCMQM, IBM MQ-Job verfolgen
- Transaktionsbefehle
  - RSVMQMTRN, IBM MQ-Transaktion auflösen
- Auslösermonitorbefehle
  - STRMQMTRM, Auslösemonitor starten
- IBM MQSC-Befehle
  - RUNMQSC, IBM MQSC-Befehle ausführen
  - STRMQMMQSC, IBM MQSC-Befehle starten

Gruppe 2

Der Rest der Befehle, für die zwei Berechtigungsstufen erforderlich sind:

1. IBM i-Berechtigung zum Ausführen des Befehls. Ein IBM MQ-Administrator legt diese Berechtigung mit dem Befehl **GRTOBJAUT** fest, durch den die Einschränkung *PUBLIC(*EXCLUDE) für einen Benutzer oder eine Benutzergruppe überschrieben wird.

   Beispiel:

   ```
   GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
   ```

2. IBM MQ-Berechtigung zum Bearbeiten der IBM MQ-Objekte, die dem Befehl oder den Befehlen zugeordnet sind, wenn in Schritt 1 die korrekte IBM i-Berechtigung erteilt wurde.

   Diese Berechtigung wird durch den Benutzer mit der entsprechenden OAM-Berechtigung für die erforderliche Aktion gesteuert, der von einem IBM MQ-Administrator mit dem Befehl **GRTMQMAUT** festgelegt wird.

Beispiel:

```
GRTMQMAUT *connect authority to the queue manager + *admchg authority to
        the queue
```

Die Befehle können wie folgt gruppiert werden:

- Kanalbefehle
  - CHGMQMCHL, IBM MQ-Kanal ändern

    Dies erfordert die Berechtigung * connect für den Warteschlangenmanager und die Berechtigung * admchg für den Kanal.
  - CPYMQMCHL, IBM MQ-Kanal kopieren

    Dies erfordert * connect und * admcrt-Berechtigung für den Warteschlangenmanager, * admdsp-Berechtigung für den zu kopierenden Standardkanaltyp und * admcrt-Berechtigung für die Kanalobjektklasse.

    Wenn zum Beispiel ein Senderkanal kopiert wird, benötigt die Berechtigung * admdsp für den Kanal SYSTEM.DEF.SENDER
  - CRTMQMCHL, IBM MQ-Kanal erstellen

    Dies erfordert * connect und * admcrt-Berechtigung für den Warteschlangenmanager, * admdsp-Berechtigung für den zu erstellenden Standardkanaltyp und * admcrt-Berechtigung für die Kanalobjektklasse.

    Wenn Sie beispielsweise einen Senderkanal erstellen, benötigt die Berechtigung * admdsp für den Kanal SYSTEM.DEF.SENDER.
  - DLTMQMCHL, IBM MQ-Kanal löschen

    Dies erfordert die Berechtigung * connect für den Warteschlangenmanager und die Berechtigung * admdlt für den Kanal.
  - RSVMQMCHL, IBM MQ-Kanal auflösen

    Dies erfordert die Berechtigung * connect für den Warteschlangenmanager und die Berechtigung * ctrlx für den Kanal.
- Anzeigebefehle

  Um die DSP-Befehle zu verarbeiten, müssen Sie den Benutzer `*connect` und die Berechtigung `*admdsp` für den Warteschlangenmanager zusammen mit allen aufgelisteten spezifischen Optionen erteilen:
  - DSPMQM, Nachrichten-WS-Manager anzeigen
  - DSPMQMAUT, IBM MQ-Objektberechtigung anzeigen
  - DSPMQMAUTI, IBM MQ-Authentifizierungsinformationen anzeigen - `*admdsp` für das Authentifizierungsdatenobjekt
  - DSPMQMCHL, IBM MQ-Kanal anzeigen - `*admdsp` für den Kanal
  - DSPMQMCSVR, IBM MQ-Befehlsserver anzeige
  - DSPMQMNL, IBM MQ-Namensliste anzeigen - `*admdsp` für die Namensliste
  - DSPMQMOBJN, IBM MQ-Objektnamen anzeigen
  - DSPMQMPRC, IBM MQ-Prozess anzeigen - `*admdsp` für den Prozess
  - DSPMQMQ, IBM MQ-Warteschlange anzeigen - `*admdsp` für de Warteschlange
  - DSPMQMTOP, IBM MQ-Thema anzeigen - `*admdsp` für das Thema
- Mit Befehlen arbeiten

Um die WRK-Befehle zu verarbeiten und die Anzeige "Optionen" aufzurufen, müssen Sie dem Warteschlangenmanager die Berechtigung *connect und *admdsp sowie alle aufgelisteten spezifischen Optionen erteilen:

- WRKMQM, Mit Nachrichten-WS-Managern arbeiten
- WRKMQMAUT, Mit IBM MQ-Objektberechtigung arbeiten
- WRKMQMAUTD, Mit IBM MQ-Objektberechtigungsdaten arbeiten
- WRKMQMAUTI, Mit IBM MQ-Authentifizierungsinformationen arbeiten
  - *admchg für den Befehl zum Ändern des IBM MQ-Authentifizierungsdatenobjekts.
  - *admcrt für den Befehl zum Erstellen und Kopieren des IBM MQ-Authentifizierungsdatenobjekts.
  - *admdlt für den Befehl zum Löschen des IBM MQ-Authentifizierungsdatenobjekts.
  - *admdsp für den Befehl zum Anzeigen des IBM MQ-Authentifizierungsdatenobjekts.
- WRKMQMCHL, mit IBM MQ-Kanal arbeiten

  Dies erfordert die folgenden Berechtigungen:
  - *admchg für den Befehl zum Ändern des IBM MQ-Kanals.
  - *admclr für den Befehl zum Abwählen des IBM MQ-Kanals.
  - *admcrt für den Befehl zum Erstellen und Kopieren des IBM MQ-Kanals.
  - *admdlt für den Befehl zum Löschen des IBM MQ-Kanals.
  - *admdsp für den Befehl zum Anzeigen des IBM MQ-Kanals.
  - *ctrl für den Befehl zum Starten des IBM MQ-Kanals.
  - *ctrl für den Befehl zum Beenden des IBM MQ-Kanals.
  - *ctrl für den Befehl zum Absetzen eines Pingsignals für den IBM MQ-Kanal.
  - *ctrlx für den Befehl zum Zurücksetzen des IBM MQ-Kanals.
  - *ctrlx für den Befehl zum Auflösen des IBM MQ-Kanals.
- WRKMQMCHST, mit IBM MQ-Kanalstatus arbeiten

  Dies erfordert die Berechtigung *admdsp für den Kanal.
- WRKMQMCL, mit IBM MQ-Clustern arbeiten
- WRKMQMCLQ, mit IBM MQ-Clusterwarteschlangen arbeiten
- WRKMQMCLQM, mit IBM MQ-Clusterwarteschlangenmanagern arbeiten
- WRKMQMLSR, mit IBM MQ-Listenern arbeiten
- WRKMQMMSG, mit IBM MQ-Nachrichten arbeiten

  Dies erfordert die Berechtigung *browse für die Warteschlange.
- WRKMQMNL, mit IBM MQ-Namenslisten arbeiten

  Dies erfordert die folgenden Berechtigungen:
  - *admchg für den Befehl zum Ändern der IBM MQ-Namensliste.
  - *admcrt für den Befehl zum Erstellen und Kopieren der IBM MQ-Namensliste.
  - *admdlt für den Befehl zum Löschen der IBM MQ-Namensliste.
  - *admdsp für den Befehl zum Anzeigen der IBM MQ-Namensliste.
- WRKMQMPRC, mit IBM MQ-Prozessen arbeiten

  Dies erfordert die folgenden Berechtigungen:
  - *admchg für den Befehl zum Ändern des IBM MQ-Prozesses.
  - *admcrt für den Befehl zum Erstellen und Kopieren des IBM MQ-Prozesses.
  - *admdlt für den Befehl zum Löschen des IBM MQ-Prozesses.

- *admdsp für den Befehl zum Anzeigen des IBM MQ-Prozesses.

– WRKMQMQ, mit IBM MQ-Warteschlangen arbeiten

  Dies erfordert die folgenden Berechtigungen:

  - *admchg für den Befehl zum Ändern der IBM MQ-Warteschlange.
  - *admclr für den Befehl zum Abwählen der IBM MQ-Warteschlange.
  - *admcrt für den Befehl zum Erstellen und Kopieren vonIBM MQ -Warteschlangen
  - *admdlt für den Befehl zum Löschen der IBM MQ-Warteschlange.
  - *admdsp für den Befehl zum Anzeigen der IBM MQ-Warteschlange.

– WRKMQMQSTS, mit IBM MQ-Warteschlangenstatus arbeiten

– WRKMQMTOP, mit IBM MQ-Themen arbeiten

  Dies erfordert die folgenden Berechtigungen:

  - *admchg für den Befehl zum Ändern des IBM MQ-Themas.
  - *admcrt für den Befehl zum Erstellen und Kopieren des IBM MQ-Themas.
  - *admdlt für den Befehl zum Löschen des IBM MQ-Themas.
  - *admdsp für den Befehl zum Anzeigen des IBM MQ-Themas.

– WRKMQMSUB, mit IBM MQ-Subskriptionen arbeiten

• Andere Kanalbefehle

  Um die Kanalbefehle zu verarbeiten, müssen Sie dem Benutzer die folgenden spezifischen Berechtigungen erteilen:

  – ENDMQMCHL, IBM MQ-Kanal beenden

    Dies erfordert die Berechtigung *connect für den Warteschlangenmanager und die Berechtigung *allmqi für die Übertragungswarteschlange, die dem Kanal zugeordnet ist.

  – ENDMQMLSR, IBM MQ-Listener beenden

    Dies erfordert die Berechtigung *connect für den Warteschlangenmanager und die *ctrl -Berechtigung für das benannte Empfangsprogrammobjekt.

  – PNGMQMCHL, Pingsignal für IBM MQ-Kanal absetzen

    Dies erfordert die Berechtigung *connect und *inq für den Warteschlangenmanager und die *ctrl -Berechtigung für das Kanalobjekt.

  – RSTMQMCHL, IBM MQ-Kanal zurücksetzen

    Dies erfordert die Berechtigung *connect für den Warteschlangenmanager.

  – STRMQMCHL, IBM MQ-Kanal starten

    Dies erfordert die Berechtigung *connect für den Warteschlangenmanager und die *ctrl -Berechtigung für das Kanalobjekt.

  – STRMQMCHLI, IBM MQ-Kanalinitiator starten

    Dies erfordert die Berechtigung *connect und *inq für den Warteschlangenmanager und die Berechtigung *allmqi für die Initialisierungswarteschlange, die der Übertragungswarteschlange des Kanals zugeordnet ist.

  – STRMQMLSR, IBM MQ-Listener starten

    Hierzu ist die Berechtigung * connect für den Warteschlangenmanager und die Berechtigung * ctrl für das benannte Empfangsprogrammobjekt erforderlich.

• Andere Befehle:

  Um die folgenden Befehle verarbeiten zu können, müssen Sie dem Benutzer die aufgelisteten spezifischen Berechtigungen erteilen:

  – CCTMQM, Verbindung zum Nachrichtenwarteschlangenmanager herstellen

Hierfür ist die IBM MQ-Objektberechtigung nicht erforderlich.

– CHGMQM, Nachrichten-WS-Manager ändern

  Dies erfordert die Berechtigung `*connect` und `*admchg` für den WS-Manager.

– CHGMQMAUTI, IBM MQ-Authentifizierungsinformationen ändern

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die Berechtigung `*admchg` und die Berechtigung `*admdsp` für das Authentifizierungsinformationsobjekt.

– CHGMQMNL, IBM MQ-Namensliste ändern

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die `*admchg` -Berechtigung für die Namensliste.

– CHGMQMPRC, IBM MQ-Prozess ändern

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die `*admchg` -Berechtigung für den Prozess.

– CHGMQMQ, IBM MQ-Warteschlange ändern

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die Berechtigung `*admchg` für die Warteschlange.

– CLRMQMQ, IBM MQ-Warteschlange abwählen

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die Berechtigung `*admclr` für die Warteschlange.

– CPYMQMAUTI, IBM MQ-Authentifizierungsinformationen kopieren

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die `*admdsp` -Berechtigung für das Authentifizierungsinformationsobjekt und die Berechtigung `*admcrt` für die Authentifizierungsinformationsobjektklasse.

– CPYMQMNL, IBM MQ-Namensliste kopieren

  Dies erfordert die Berechtigung `*connect` und `*admcrt` für den WS-Manager.

– CPYMQMPRC, IBM MQ-Prozess kopieren

  Dies erfordert die Berechtigung `*connect` und `*admcrt` für den WS-Manager.

– CPYMQMQ, IBM MQ-Warteschlange kopieren

  Dies erfordert die Berechtigung `*connect` und `*admcrt` für den WS-Manager.

– CRTMQMAUTI, IBM MQ-Authentifizierungsinformationen erstellen

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die `*admdsp` -Berechtigung für das Authentifizierungsinformationsobjekt und die Berechtigung `*admcrt` für die Authentifizierungsinformationsobjektklasse.

– CRTMQMNL, IBM MQ-Namensliste erstellen

  Dies erfordert die Berechtigung `*connect` und `*admcrt` für den Warteschlangenmanager und die `*admdsp` -Berechtigung für die Standardnamensliste.

– CRTMQMPRC, IBM MQ-Prozess erstellen

  Dies erfordert die Berechtigung `*connect` und `*admcrt` für den Warteschlangenmanager und die Berechtigung `*admdsp` für den Standardprozess.

– CRTMQMQ, IBM MQ-Warteschlange erstellen

  Dies erfordert die Berechtigung `*connect` und `*admcrt` für den Warteschlangenmanager und die Berechtigung `*admdsp` für die Standardwarteschlange.

– CVTMQMDTA, IBM MQ-Datentypbefehl umwandeln

  Hierfür ist die IBM MQ-Objektberechtigung nicht erforderlich.

– DLTMQMAUTI, IBM MQ-Authentifizierungsinformationen löschen

Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die `*ctrlx` -Berechtigung für das Authentifizierungsinformationsobjekt.

- DLTMQMNL, IBM MQ-Namensliste löschen

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die `*admdlt` -Berechtigung für die Namensliste.

- DLTMQMPRC, IBM MQ-Prozess löschen

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die `*admdlt` -Berechtigung für den Prozess.

- DLTMQMQ, IBM MQ-Warteschlange löschen

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager und die Berechtigung `*admdlt` für die Warteschlange.

- DSCMQM, Verbindung zum Nachrichten-WS-Manager trennen

  Hierfür ist die IBM MQ-Objektberechtigung nicht erforderlich.

- RFRMQMAUT, Sicherheit aktualisieren

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager.

- RFRMQMCL, Cluster aktualisieren

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager.

- RSMMQMCLQM, Clusterwarteschlangenmanager wiederaufnehmen

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager.

- RSTMQMCL, Cluster zurücksetzen

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager.

- SPDMQMCLQM, Clusterwarteschlangenmanager aussetzen

  Dies erfordert die Berechtigung `*connect` für den Warteschlangenmanager.

IBM i *Zugriffsberechtigungen für IBM i*

Verwenden Sie diese Informationen, um die Zugriffsberechtigungsbefehle zu verstehen.

Berechtigungen, die durch das Schlüsselwort AUT in den Befehlen `GRTMQMAUT` und `RVKMQMAUT` definiert werden, können wie folgt kategorisiert werden:

- Autorisierungen im Zusammenhang mit MQI-Aufrufen
- Berechtigungsbezogene Verwaltungsbefehle
- Kontextberechtigungen
- Allgemeine Berechtigungen, d. B. für MQI-Aufrufe, für Befehle oder beides

In den folgenden Tabellen werden die verschiedenen Berechtigungen mit Hilfe des Parameters AUT für MQI-Aufrufe, Kontextaufrufe, MQSC-und PCF-Befehle sowie generische Operationen aufgelistet.

Tabelle 15. Berechtigungen für MQI-Aufrufe

AUT	**Beschreibung**
`*ALTUSR`	Erlauben Sie, dass die Berechtigung eines anderen Benutzers für MQOPEN-und MQPUT1-Aufrufe verwendet wird.
`*BROWSE`	Eine Nachricht aus einer Warteschlange über einen MQGET-Aufruf mit der Option BROWSE abrufen.
`*CONNECT`	Die Anwendung mit dem angegebenen Warteschlangenmanager über einen MQCONN-Aufruf verbinden.
`*GET`	Eine Nachricht aus einer Warteschlange über einen MQGET-Aufruf abrufen.

Tabelle 15. Berechtigungen für MQI-Aufrufe (Forts.)	
AUT	**Beschreibung**
*INQ	Erstellen Sie eine Abfrage für eine bestimmte Warteschlange, indem Sie einen MQINQ-Aufruf absetzen.
*PUB	Öffnen Sie ein Thema, um eine Nachricht unter Verwendung eines MQPUT-Aufrufs zu veröffentlichen.
*PUT	Schreiben Sie eine Nachricht in eine bestimmte Warteschlange, indem Sie einen MQPUT-Aufruf absetzen.
*RESUME	Wiederaufnehmen einer Subskription mit einem MQSUB-Aufruf.
*SET	Sie können Attribute in einer Warteschlange aus dem MQI festlegen, indem Sie einen MQSET-Aufruf absetzen. Wenn Sie eine Warteschlange für mehrere Optionen öffnen, müssen Sie für jeden dieser Optionen berechtigt sein.
*SUB	Erstellen, Ändern oder Fortsetzen einer Subskription für ein Thema unter Verwendung eines MQSUB-Aufrufs.

Tabelle 16. Berechtigungen für Kontextaufrufe	
AUT	**Beschreibung**
*PASSALL	Übergeben Sie den gesamten Kontext in der angegebenen Warteschlange. Alle Kontextfelder werden aus der ursprünglichen Anforderung kopiert.
*PASSID	Kennungskontext in der angegebenen Warteschlange übergeben. Der Identitätskontext stimmt mit dem Kontext der Anforderung überein.
*SETALL	Legen Sie den gesamten Kontext in der angegebenen Warteschlange fest. Dies wird von speziellen Systemdienstprogrammen verwendet.
*SETID	Legen Sie den Identitätskontext in der angegebenen Warteschlange fest. Dies wird von speziellen Systemdienstprogrammen verwendet.

Tabelle 17. Berechtigungen für MQSC-und PCF-Aufrufe	
AUT	**Beschreibung**
*ADMCHG	Ändern Sie die Attribute des angegebenen Objekts.
*ADMCLR	Löschen Sie das angegebene Objekt (nur Objektbefehl PCF Clear object).
*ADMCRT	Erstellen Sie Objekte des angegebenen Typs.
*ADMDLT	Das angegebene Objekt löschen.
*ADMDSP	Zeigt die Attribute des angegebenen Objekts an.

Tabelle 18. Berechtigungen für generische Operationen	
AUT	**Beschreibung**
*ALL	Verwenden Sie alle Operationen, die für das Objekt gelten. Die Berechtigung `all` entspricht der Verknüpfung der für den Objekttyp relevanten Berechtigungen `alladm`, `allmqi` und `system`.
*ALLADM	Führen Sie alle Verwaltungsoperationen aus, die auf das Objekt anwendbar sind.
*ALLMQI	Verwenden Sie alle MQI-Aufrufe, die auf das Objekt anwendbar sind.

Tabelle 18. Berechtigungen für generische Operationen (Forts.)	
AUT	**Beschreibung**
*CTRL	Steuerung des Systemstarts und -abschlusses von Kanälen, Empfangsprogrammen und Services.
*CTRLX	Folgenummer zurücksetzen und unbestätigte Kanäle auflösen.

IBM i *Zugriffsberechtigungsbefehle unter IBM i verwenden*

Verwenden Sie diese Informationen, um Informationen zu den Zugriffsberechtigungsbefehlen zu erhalten, und verwenden Sie die Befehlsbeispiele.

Befehl GRTMQMAUT verwenden

Wenn Sie über die erforderliche Berechtigung verfügen, können Sie den Befehl GRTMQMAUT verwenden, um die Berechtigung eines Benutzerprofils oder einer Benutzergruppe zu erteilen, um auf ein bestimmtes Objekt zuzugreifen. Die folgenden Beispiele zeigen, wie der Befehl GRTMQMAUT verwendet wird:

1. ```
   GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
             AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
   ```

   In diesem Beispiel gilt Folgendes:

   - RED.LOCAL.QUEUE ist der Objektname.
   - *LCLQ (lokale Warteschlange) ist der Objekttyp.
   - GROUPA ist der Name eines Benutzerprofils auf dem System, für das die Berechtigungen geändert werden sollen. Dieses Profil kann als Gruppenprofil für andere Benutzer verwendet werden.
   - *BROWSE und *PUT sind die Berechtigungen, die der angegebenen Warteschlange erteilt werden.

     *BROWSE fügt die Berechtigung zum Durchsuchen von Nachrichten in der Warteschlange hinzu (um MQGET mit der Suchoption auszugeben).

     *PUT fügt die Berechtigung zum put (MQPUT) -Nachrichten in die Warteschlange hinzu.
   - saturn.queue.manager ist der Name des Warteschlangenmanagers.
2. Der folgende Befehl erteilt Benutzern JACK und JILL alle gültigen Berechtigungen für alle Prozessdefinitionen für den Standardwarteschlangenmanager.

   ```
   GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
   ```
3. Der folgende Befehl erteilt dem Benutzer GEORGE die Berechtigung, eine Nachricht in die Warteschlange ORDERS auf den Warteschlangenmanager TRENT zu stellen.

   ```
   GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
   GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
   ```

Befehl RVKMQMAUT verwenden

Wenn Sie über die erforderliche Berechtigung verfügen, können Sie den Befehl RVKMQMAUT verwenden, um zuvor erteilte Berechtigungen eines Benutzerprofils oder einer Benutzergruppe zu entfernen, um auf ein bestimmtes Objekt zuzugreifen. Die folgenden Beispiele zeigen, wie der Befehl RVKMQMAUT verwendet wird:

1. ```
   RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
   AUT(*PUT) MQMNAME('saturn.queue.manager')
   ```

Die Berechtigung zum Angeben von Nachrichten an die angegebene Warteschlange, die im vorherigen Beispiel erteilt wurde, wird für GROUPA entfernt.

2.
```
RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
MQMNAME(PAYROLLQM)
```

Die Berechtigung zum Abrufen von Nachrichten aus allen Warteschlangen mit einem Namen, der mit den Zeichen PAY beginnt und deren Eigner der Warteschlangenmanager PAYROLLQM ist, wird von allen Benutzern des Systems entfernt, es sei denn, sie oder eine Gruppe, zu der sie gehören, wurden separat autorisiert.

Befehl DSPMQMAUT verwenden

Der Befehl DSPMQMAUT (MQM-Berechtigung anzeigen) zeigt für das angegebene Objekt und den Benutzer die Liste der Berechtigungen an, die der Benutzer für das Objekt hat. Das folgende Beispiel zeigt, wie der Befehl verwendet wird:

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

Verwendung des Befehls RFRMQMAUT

Mit dem Befehl RFRMQMAUT (MQM-Sicherheit aktualisieren) können Sie die Berechtigungsgruppeninformationen des OAM sofort aktualisieren und die Änderungen auf Betriebssystemebene widerspiegeln, ohne dass der WS-Manager gestoppt und neu gestartet werden muss. Das folgende Beispiel zeigt, wie der Befehl verwendet wird:

```
RFRMQMAUT MQMNAME(ADMINQM)
```

IBM i Tabelle mit Berechtigungsspezifikationen für IBM i

Anhand dieser Informationen können Sie feststellen, welche Berechtigung erforderlich ist, um bestimmte API-Aufrufe und bestimmte Optionen für diese Aufrufe in Warteschlangenobjekten, Prozessobjekten und WS-Manager-Objekten zu verwenden.

Die Berechtigungsspezifikationstabellen ab Tabelle Tabelle 19 auf Seite 180 stellen genau dar, wie die einzelnen Berechtigungen funktionieren und welche Einschränkungen für sie gelten. Die Tabellen gelten für die folgenden Situationen:

- Anwendungen, die MQI-Aufrufe absetzen
- Verwaltungsprogramme, die MQSC-Befehle als Escape-PCFs ausgeben
- Verwaltungsprogramme, die PCF-Befehle absetzen

In diesem Abschnitt werden die Informationen in Form einer Gruppe von Tabellen dargestellt, die die folgenden Daten angeben:

Aktion, die ausgeführt werden soll
MQI-Option, MQSC-Befehl oder PCF-Befehl.

Zugriffssteuerungsobjekt
Warteschlange, Prozessdefinition, Warteschlangenmanager, Namensliste, Kanal, Clientverbindungskanal, Listener-, Service-oder Authentifizierungsinformationsobjekt.

Erforderliche Berechtigung
Als MQZAO_-Konstante ausgedrückt.

In den Tabellen entsprechen die Konstanten mit dem Präfix MQZAO_ den Schlüsselwörtern in der Berechtigungsliste für die Befehle **GRTMQMAUT** und **RVKMQMAUT** für die jeweilige Entität. Beispiel: MQZAO_BROWSE entspricht dem Schlüsselwort *BROWSE . Ebenso entspricht das Schlüsselwort

MQZAO_SET_ALL_CONTEXT dem Schlüsselwort *SETALL usw. Diese Konstanten werden in der Headerdatei cmqzc.h definiert, die im Lieferumfang des Produkts enthalten ist.

MQI-Berechtigungen

Eine Anwendung darf bestimmte MQI-Aufrufe und -Optionen nur dann absetzen, wenn die Benutzer-ID, unter der sie ausgeführt wird (oder deren Berechtigungen vorausgesetzt werden können), die entsprechende Berechtigung erteilt hat.

Für vier MQI-Aufrufe sind Berechtigungsprüfungen erforderlich: MQCONN, MQOPEN, MQPUT1 und MQCLOSE.

Bei MQOPEN und MQPUT1 wird die Berechtigungs-Prüfung auf den Namen des zu öffnende Objekts, nicht auf den Namen oder die Namen, die sich nach dem Namen eines Namens ergeben, durchgeführt. Beispielsweise kann einer Anwendung die Berechtigung zum Öffnen einer Aliaswarteschlange erteilt werden, ohne dass die Berechtigung zum Öffnen der Basiswarteschlange, in die der Aliasname aufgelöst wird, geöffnet werden kann. Die Regel ist, dass die Prüfung bei der ersten Definition ausgeführt wird, die während des Prozesses der Namensauflösung auftritt, der kein WS-Manager-Aliasname ist, es sei denn, die Aliasdefinition des Warteschlangenmanagers wird direkt geöffnet. Das heißt, ihr Name wird im Feld *ObjectName* des Objektdeskriptors angezeigt. Die Berechtigung wird für das jeweilige Objekt, das gerade geöffnet wird, immer benötigt. In einigen Fällen ist eine zusätzliche warteschlangenunabhängige Berechtigung erforderlich, die über eine Berechtigung für das WS-Manager-Objekt ermittelt wird.

In Tabelle 19 auf Seite 180, Tabelle 20 auf Seite 180, Tabelle 21 auf Seite 181 und Tabelle 22 auf Seite 182 sind die für die einzelnen Aufrufe erforderlichen Berechtigungen zusammengestellt.

Anmerkung: In diesen Tabellen werden Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services oder Authentifizierungsinformationsobjekte nicht erwähnt. Dies liegt daran, dass keine der Berechtigungen für diese Objekte gilt, mit Ausnahme von MQOO_INQUIRE, für die die gleichen Berechtigungen wie für die anderen Objekte gelten.

Tabelle 19. Für MQCONN-Aufrufe erforderliche Sicherheitsberechtigung

Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 182)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQCONN, Option	Nicht zutreffend	Nicht zutreffend	MQZAO_CONNECT

Tabelle 20. Für MQOPEN-Aufrufe erforderliche Sicherheitsberechtigung

Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 182)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQOO_INQUIRE	MQZAO_INQUIRE **(„2” auf Seite 182)**	MQZAO_INQUIRE **(„2” auf Seite 182)**	MQZAO_INQUIRE **(„2” auf Seite 182)**
MQOO_BROWSE	MQZAO_BROWSE	Nicht zutreffend	Keine Prüfung
MQOO_INPUT_*	MQZAO_INPUT	Nicht zutreffend	Keine Prüfung
MQOO_SAVE_ ALL_CONTEXT **(„3” auf Seite 182)**	MQZAO_INPUT	Nicht zutreffend	Nicht zutreffend
MQOO_OUTPUT (normale Warteschlange) **(„4” auf Seite 182)**	MQZAO_OUTPUT	Nicht zutreffend	Nicht zutreffend
MQOO_PASS_ IDENTITY_CONTEXT **(„5” auf Seite 182)**	MQZAO_PASS_ IDENTITY_CONTEXT	Nicht zutreffend	Keine Prüfung

Tabelle 20. Für MQOPEN-Aufrufe erforderliche Sicherheitsberechtigung (Forts.)			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 182)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQOO_PASS_ALL_ CONTEXT **(„5” auf Seite 182, „6” auf Seite 182)**	MQZAO_PASS _ALL_CONTEXT	Nicht zutreffend	Keine Prüfung
MQOO_SET_ IDENTITY_CONTEXT **(„5” auf Seite 182, „6” auf Seite 182)**	MQZAO_SET_ IDENTITY_CONTEXT	Nicht zutreffend	MQZAO_SET_ IDENTITY_CONTEXT **(„7” auf Seite 182)**
MQOO_SET_ ALL_CONTEXT **(„5” auf Seite 182, „8” auf Seite 182)**	MQZAO_SET_ ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ ALL_CONTEXT **(„7” auf Seite 182)**
MQOO_OUTPUT (Übertragungswarteschlange) **(„9” auf Seite 182)**	MQZAO_SET_ ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ ALL_CONTEXT **(„7” auf Seite 182)**
MQOO_SET	MQZAO_SET	Nicht zutreffend	Keine Prüfung
MQOO_ALTERNATE_ USER_AUTHORITY	**(„10” auf Seite 182)**	**(„10” auf Seite 182)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(„10” auf Seite 182, „11” auf Seite 182)**

Tabelle 21. Für MQPUT1-Aufrufe erforderliche Sicherheitsberechtigung			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 182)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(„12” auf Seite 182)**	Nicht zutreffend	Keine Prüfung
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(„12” auf Seite 182)**	Nicht zutreffend	Keine Prüfung
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(„12” auf Seite 182)**	Nicht zutreffend	MQZAO_SET_ IDENTITY_CONTEXT **(„7” auf Seite 182)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(„12” auf Seite 182)**	Nicht zutreffend	MQZAO_SET_ ALL_CONTEXT **(„7” auf Seite 182)**
(Übertragungswarteschlange) **(„9” auf Seite 182)**	MQZAO_SET_ ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ ALL_CONTEXT **(„7” auf Seite 182)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(„13” auf Seite 182)**	Nicht zutreffend	MQZAO_ALTERNATE_ USER_AUTHORITY **(„11” auf Seite 182)**

Tabelle 22. Für MQCLOSE-Aufrufe erforderliche Sicherheitsberechtigung			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 182)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQCO_DELETE	MQZAO_DELETE **(„14” auf Seite 182)**	Nicht zutreffend	Nicht zutreffend
MQCO_DELETE _PURGE	MQZAO_DELETE **(„14” auf Seite 182)**	Nicht zutreffend	Nicht zutreffend

Hinweise zu den Tabellen:

1. Wenn eine Modellwarteschlange geöffnet wird:
   - Die Berechtigung MQZAO_DISPLAY wird für die Modellwarteschlange zusätzlich zur Berechtigung zum Öffnen der Modellwarteschlange für den Typ des Zugriffs, für den Sie geöffnet werden, benötigt.
   - Die Berechtigung MQZAO_CREATE ist nicht erforderlich, um die dynamische Warteschlange zu erstellen.
   - Die Benutzer-ID, die zum Öffnen der Modellwarteschlange verwendet wird, wird automatisch allen warteschlangenspezifischen Berechtigungen (äquivalent zu MQZAO_ALL) für die erstellte dynamische Warteschlange erteilt.
2. Abhängig vom Typ des Objekts, das geöffnet wird, wird entweder die Warteschlange, der Prozess, die Namensliste oder das Warteschlangenmanagerobjekt überprüft.
3. MQOO_INPUT_ * muss ebenfalls angegeben werden. Diese Option ist für eine lokale, eine Modell- oder eine Aliaswarteschlange gültig.
4. Diese Prüfung wird für alle ausgehenden Fälle, mit Ausnahme des in Anmerkung „9” auf Seite 182 genannten Falls, ausgeführt.
5. MQOO_OUTPUT muss ebenfalls angegeben werden.
6. MQOO_PASS_IDENTITY_CONTEXT wird auch von dieser Option impliziert.
7. Diese Berechtigung ist sowohl für das Warteschlangenmanagerobjekt als auch für die bestimmte Warteschlange erforderlich.
8. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT und MQOO_SET_IDENTITY_CONTEXT werden ebenfalls von dieser Option impliziert.
9. Diese Prüfung wird für eine lokale oder Modellwarteschlange ausgeführt, die über ein *Usage* -Warteschlangenattribut von MQUS_TRANSMISSION verfügt und direkt für die Ausgabe geöffnet wird. Sie findet keine Anwendung, wenn eine ferne Warteschlange geöffnet wird (entweder durch Angabe der Namen des fernen Warteschlangenmanagers und der fernen Warteschlange oder durch Angabe des Namens einer lokalen Definition der fernen Warteschlange).
10. Es muss mindestens eine von MQOO_INQUIRE (für einen beliebigen Objekttyp) oder (für Warteschlangen) MQOO_BROWSE, MQOO_INPUT_ *, MQOO_OUTPUT oder MQOO_SET angegeben werden. Die durchgeführte Prüfung erfolgt wie bei den anderen angegebenen Optionen unter Verwendung der angegebenen alternativen Benutzer-ID für die spezielle Objektberechtigung und der aktuellen Anwendungsberechtigung für die Prüfung MQZAO_ALTERNATE_USER_IDENTIFIER.
11. Mit dieser Berechtigung kann jede beliebige *AlternateUserId* angegeben werden.
12. Es wird auch eine MQZAO_OUTPUT-Prüfung durchgeführt, wenn die Warteschlange kein Warteschlangenattribut *Usage* von MQUS_TRANSMISSION hat.
13. Die durchgeführte Prüfung erfolgt wie bei den anderen angegebenen Optionen unter Verwendung der angegebenen alternativen Benutzer-ID für die benannte Warteschlangenberechtigung und der aktuellen Anwendungsberechtigung für die Prüfung MQZAO_ALTERNATE_USER_IDENTIFIER.
14. Die Prüfung wird nur durchgeführt, wenn beide der folgenden Aussagen wahr sind:
    - Eine permanente dynamische Warteschlange wird geschlossen und gelöscht.

- Die Warteschlange wurde nicht von dem MQOPEN-Befehl erstellt, der die verwendete Objektkennung zurückgegeben hat.

Sonst gibt es keine Prüfung.

Allgemeine Hinweise:

1. Die Sonderberechtigung MQZAO_ALL_MQI enthält alle folgenden Berechtigungen, die für den Objekttyp relevant sind:
   - MQZAO_CONNECT
   - MQZAO_INQUIRE
   - MQZAO_SET
   - MQZAO_BROWSE
   - MQZAO_INPUT
   - MQZAO_OUTPUT
   - MQZAO_PASS_IDENTITY_CONTEXT
   - MQZAO_PASS_ALL_CONTEXT
   - MQZAO_SET_IDENTITY_CONTEXT
   - MQZAO_SET_ALL_CONTEXT
   - MQZAO_ALTERNATE_USER_AUTHORITY
2. MQZAO_DELETE (siehe Anmerkung „14" auf Seite 182) und MQZAO_DISPLAY gelten als Verwaltungsberechtigungen. Sie sind daher nicht in MQZAO_ALL_MQI enthalten.
3. *Keine Prüfung* bedeutet, dass keine Berechtigungsprüfung durchgeführt wird.
4. *Nicht zutreffend* bedeutet, dass die Berechtigungsprüfung für diese Operation nicht relevant ist. Sie können beispielsweise keinen MQPUT-Aufruf an ein Prozessobjekt ausgeben.

IBM i *Berechtigungen für MQSC-Befehle in Escape-PCFs unter IBM i*

Mit diesen Berechtigungen kann ein Benutzer Verwaltungsbefehle als Escape-PCF-Nachricht ausgeben. Diese Methoden ermöglichen es einem Programm, einen Verwaltungsbefehl als Nachricht an einen Warteschlangenmanager zu senden, um für diesen Benutzer ausgeführt zu werden.

In diesem Abschnitt werden die Berechtigungen zusammengefasst, die für die einzelnen in Escape PCF enthaltenen MQSC-Befehle erforderlich sind.

Nicht zutreffend bedeutet, dass die Berechtigungsprüfung für diese Operation nicht relevant ist.

Die Benutzer-ID, unter der das Programm, das den Befehl übergibt, ausgeführt wird, muss außerdem über die folgenden Berechtigungen verfügen:

- Berechtigung MQZAO_CONNECT für den WS-Manager
- Berechtigung DISPLAY auf dem WS-Manager zur Ausführung von PCF-Befehlen
- Berechtigung zum Absetzen der MQSC-Befehle im Text des Escape-PCF-Befehls

ALTER *object*

Objekt	**Erforderliche Berechtigung**
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	MQZAO_ÄNDERUNG
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsdaten	MQZAO_ÄNDERUNG

Objekt	Erforderliche Berechtigung
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Empfangsprogramm	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG

CLEAR *object*

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_CLEAR
Thema	MQZAO_CLEAR
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	Nicht zutreffend
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

DEFINE *Objekt* NOREPLACE („1” auf Seite 187)

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(„2” auf Seite 187)**
Thema	MQZAO_CREATE **(„2” auf Seite 187)**
Prozess	MQZAO_CREATE **(„2” auf Seite 187)**
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_CREATE **(„2” auf Seite 187)**
Authentifizierungsdaten	MQZAO_CREATE **(„2” auf Seite 187)**
Kanal	MQZAO_CREATE **(„2” auf Seite 187)**
Clientverbindungskanal	MQZAO_CREATE **(„2” auf Seite 187)**
Empfangsprogramm	MQZAO_CREATE **(„2” auf Seite 187)**
Service	MQZAO_CREATE **(„2” auf Seite 187)**

DEFINE *object* REPLACE („1” auf Seite 187, „3” auf Seite 187)

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	Nicht zutreffend

Objekt	Erforderliche Berechtigung
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsdaten	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Empfangsprogramm	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG

DELETE *object*

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DELETE
Thema	MQZAO_DELETE
Prozess	MQZAO_DELETE
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_DELETE
Authentifizierungsdaten	MQZAO_DELETE
Kanal	MQZAO_DELETE
Clientverbindungskanal	MQZAO_DELETE
Empfangsprogramm	MQZAO_DELETE
Service	MQZAO_DELETE

DISPLAY *object*

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DISPLAY
Thema	MQZAO_DISPLAY
Prozess	MQZAO_DISPLAY
Warteschlangenmanager	MQZAO_DISPLAY
Namensliste	MQZAO_DISPLAY
Authentifizierungsdaten	MQZAO_DISPLAY
Kanal	MQZAO_DISPLAY
Clientverbindungskanal	MQZAO_DISPLAY
Empfangsprogramm	
Service	

PING CHANNEL

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend

Objekt	Erforderliche Berechtigung
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

RESET CHANNEL

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL_EXTENDED
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

GELÖST-CHANNEL

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL_EXTENDED
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

START *object*

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	MQZAO_CONTROL
Service	MQZAO_CONTROL

STOP *object*

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	MQZAO_CONTROL
Service	MQZAO_CONTROL

Anmerkung:

1. Bei DEFINE-Befehlen wird die Berechtigung MQZAO_DISPLAY auch für das LIKE-Objekt benötigt, wenn ein Objekt angegeben wird, oder auf dem entsprechenden Objekt SYSTEM.DEFAULT.xxx, wenn LIKE weggelassen wird.
2. Die Berechtigung MQZAO_CREATE ist nicht spezifisch für ein bestimmtes Objekt oder einen bestimmten Objekttyp. Die Erstellungsberechtigung wird für alle Objekte für einen bestimmten Warteschlangenmanager erteilt, indem der Objekttyp QMGR im Befehl GRTMQMAUT angegeben wird.
3. Diese Option gilt, wenn das zu ersetzendes Objekt bereits vorhanden ist. Ist dies nicht der Fall, ist die Prüfung wie für DEFINE *object* NOREPLACE.

IBM i *Berechtigungen für PCF-Befehle unter IBM i*

Diese Berechtigungen ermöglichen es einem Benutzer, Verwaltungsbefehle als PCF-Befehle auszugeben. Diese Methoden ermöglichen es einem Programm, einen Verwaltungsbefehl als Nachricht an einen Warteschlangenmanager zu senden, um für diesen Benutzer ausgeführt zu werden.

In diesem Abschnitt werden die Berechtigungen zusammengefasst, die für die einzelnen PCF-Befehle erforderlich sind.

Keine Prüfung bedeutet, dass keine Berechtigungsprüfung durchgeführt wird; *Nicht zutreffend* bedeutet, dass die Berechtigungsprüfung für diese Operation nicht relevant ist.

Die Benutzer-ID, unter der das Programm, das den Befehl übergibt, ausgeführt wird, muss außerdem über die folgenden Berechtigungen verfügen:

- Berechtigung MQZAO_CONNECT für den WS-Manager
- Berechtigung DISPLAY auf dem WS-Manager zur Ausführung von PCF-Befehlen

Die Sonderberechtigung MQZAO_ALL_ADMIN enthält die folgenden Berechtigungen:

- MQZAO_ÄNDERUNG
- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY
- MQZAO_CONTROL
- MQZAO_CONTROL_EXTENDED

MQZAO_CREATE ist nicht enthalten, da es nicht spezifisch für ein bestimmtes Objekt oder einen bestimmten Objekttyp ist.

Change *object*

Objekt	**Erforderliche Berechtigung**
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	MQZAO_ÄNDERUNG
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsdaten	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Empfangsprogramm	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG

Löschen Sie *object*.

Objekt	**Erforderliche Berechtigung**
Warteschlange	MQZAO_CLEAR
Thema	MQZAO_CLEAR
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	Nicht zutreffend
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend

Objekt	Erforderliche Berechtigung
Service	Nicht zutreffend

***Objekt* kopieren (ohne Ersetzen) („1” auf Seite 193)**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(„2” auf Seite 193)**
Thema	MQZAO_CREATE **(„2” auf Seite 193)**
Prozess	MQZAO_CREATE **(„2” auf Seite 193)**
Warteschlangenmanager	Nicht zutreffend
NamelistMQZAO_CREATE	MQZAO_CREATE **(„2” auf Seite 193)**
Authentifizierungsdaten	MQZAO_CREATE **(„2” auf Seite 193)**
Kanal	MQZAO_CREATE **(„2” auf Seite 193)**
Clientverbindungskanal	MQZAO_CREATE **(„2” auf Seite 193)**
Empfangsprogramm	MQZAO_CREATE **(„2” auf Seite 193)**
Service	MQZAO_CREATE **(„2” auf Seite 193)**

***object* kopieren (mit Ersetzen) („1” auf Seite 193, „4” auf Seite 193)**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsdaten	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Empfangsprogramm	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG

***Objekt* erstellen (ohne Ersetzen) („3” auf Seite 193)**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(„2” auf Seite 193)**
Thema	MQZAO_CREATE **(„2” auf Seite 193)**
Prozess	MQZAO_CREATE **(„2” auf Seite 193)**
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_CREATE **(„2” auf Seite 193)**
Authentifizierungsdaten	MQZAO_CREATE **(„2” auf Seite 193)**
Kanal	MQZAO_CREATE **(„2” auf Seite 193)**

Objekt	Erforderliche Berechtigung
Clientverbindungskanal	MQZAO_CREATE **(„2” auf Seite 193)**
Empfangsprogramm	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG

***object* erstellen (mit Ersetzen) („3” auf Seite 193, „4” auf Seite 193)**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsdaten	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Empfangsprogramm	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG

***object* löschen**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DELETE
Thema	MQZAO_DELETE
Prozess	MQZAO_DELETE
Warteschlangenmanager	MQZAO_DELETE
Namensliste	MQZAO_DELETE
Authentifizierungsdaten	MQZAO_DELETE
Kanal	MQZAO_DELETE
Clientverbindungskanal	MQZAO_DELETE
Empfangsprogramm	MQZAO_DELETE
Service	MQZAO_DELETE

Inquire *object*

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DISPLAY
Thema	MQZAO_DISPLAY
Prozess	MQZAO_DISPLAY
Warteschlangenmanager	MQZAO_DISPLAY
Namensliste	MQZAO_DISPLAY

Objekt	Erforderliche Berechtigung
Authentifizierungsdaten	MQZAO_DISPLAY
Kanal	MQZAO_DISPLAY
Clientverbindungskanal	MQZAO_DISPLAY
Empfangsprogramm	MQZAO_DISPLAY
Service	MQZAO_DISPLAY

***object* -Namen inquire**

Objekt	Erforderliche Berechtigung
Warteschlange	Keine Prüfung
Thema	Keine Prüfung
Prozess	Keine Prüfung
Warteschlangenmanager	Keine Prüfung
Namensliste	Keine Prüfung
Authentifizierungsdaten	Keine Prüfung
Kanal	Keine Prüfung
Clientverbindungskanal	Keine Prüfung
Empfangsprogramm	Keine Prüfung
Service	Keine Prüfung

Pingkanal

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

Kanal zurücksetzen

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend

Objekt	Erforderliche Berechtigung
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL_EXTENDED
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

Warteschlangenstatistik zurücksetzen

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DISPLAY und MQZAO_CHANGE
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	Nicht zutreffend
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	
Service	

Kanal auflösen

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL_EXTENDED
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

Kanal starten

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend

Objekt	Erforderliche Berechtigung
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

Kanal stoppen

Objekt	Erforderliche Berechtigung
Warteschlange	Nicht zutreffend
Thema	Nicht zutreffend
Prozess	Nicht zutreffend
Warteschlangenmanager	Nicht zutreffend
Namensliste	Nicht zutreffend
Authentifizierungsdaten	Nicht zutreffend
Kanal	MQZAO_CONTROL
Clientverbindungskanal	Nicht zutreffend
Empfangsprogramm	Nicht zutreffend
Service	Nicht zutreffend

Anmerkung:

1. Für Kopierbefehle ist auch die Berechtigung MQZAO_DISPLAY für das From-Objekt erforderlich.
2. Die Berechtigung MQZAO_CREATE ist nicht spezifisch für ein bestimmtes Objekt oder einen bestimmten Objekttyp. Die Erstellungsberechtigung wird für alle Objekte für einen bestimmten Warteschlangenmanager erteilt, indem der Objekttyp QMGR im Befehl GRTMQMAUT angegeben wird.
3. Für Erstellungsbefehle ist auch die Berechtigung MQZAO_DISPLAY für das entsprechende SYSTEM.DEFAULT.* Objekt.
4. Diese Option gilt, wenn das zu ersetzendes Objekt bereits vorhanden ist. Ist dies nicht der Fall, ist die Prüfung wie für Kopieren oder Erstellen ohne Ersetzen.

IBM i Generische OAM-Profile für IBM i

Mit generischen OAM-Profilen (Object Authority Manager) können Sie die Berechtigung eines Benutzers für viele Objekte auf einmal festlegen, anstatt separate **GRTMQMAUT** -Befehle für jedes einzelne Objekt ausgeben zu müssen, wenn es erstellt wird. Durch die Verwendung generischer Profile im Befehl **GRTMQMAUT** können Sie eine generische Berechtigung für alle zukünftigen erstellten Objekte festlegen, die diesem Profil entsprechen.

Der Rest dieses Abschnitts beschreibt die Verwendung generischer Profile im Detail:

- „Platzhalterzeichen verwenden” auf Seite 194

- „Profilprioritäten” auf Seite 194

Platzhalterzeichen verwenden

Was ein Profil generisch macht, ist die Verwendung von Sonderzeichen (Platzhalterzeichen) im Profilnamen. Beispielsweise stimmt das Platzhalterzeichen Fragezeichen (?) mit einem beliebigen einzelnen Zeichen in einem Namen überein. Wenn Sie ABC.?EF angeben, gilt die Berechtigung, die Sie diesem Profil erteilen, für alle Objekte, die mit den Namen ABC.DEF, ABC.CEF, ABC.BEF usw. erstellt wurden.

Folgende Platzhalterzeichen stehen zur Verfügung:

?

Verwenden Sie das Fragezeichen (?) anstelle eines beliebigen einzelnen Zeichens. AB.?D würde z. B. für die Objekte AB.CD, AB.ED und AB.FD gelten.

*****

Verwenden Sie den Stern (*) wie folgt:

- Ein *Qualifikationsmerkmal* in einem Profilnamen, das einem beliebigen Qualifikationsmerkmal in einem Objektnamen entspricht. Ein Qualifikationsmerkmal ist der Teil eines Objektnamens, der durch einen Punkt begrenzt wird. In ABC.DEF.GHI, beispielsweise sind die Qualifikationsmerkmale ABC, DEF und GHI.

  ABC.*.JKL würde z. B. für die Objekte ABC.DEF.JKL und ABC.GHI.JKL gelten. (Beachten Sie, dass es **nicht** für ABC.JKL gelten würde; * * verwendet in diesem Kontext immer ein Qualifikationsmerkmal.)
- Ein Zeichen in einem Qualifikationsmerkmal in einem Profilnamen, das null oder mehr Zeichen innerhalb des Qualifikationsmerkmals in einem Objektnamen entspricht.

  ABC.DE*.JKL würde z. B. für die Objekte ABC.DE.JKL, ABC.DEF.JKL und ABC.DEGH.JKL gelten.

******

Verwenden Sie den doppelten Stern (**) ***einmal*** in einem Profilnamen wie folgt:

- Der gesamte Profilname, der mit allen Objektnamen übereinstimmt. Wenn Sie zum Beispiel das Schlüsselwort OBJTYPE (*PRC) verwenden, um Prozesse zu identifizieren, verwenden Sie ** als Profilnamen, und ändern Sie die Berechtigungen für alle Prozesse.
- Als Anfangs-, Mittel-oder Endqualifikationsmerkmal in einem Profilnamen, der null oder mehr Qualifikationsmerkmale in einem Objektnamen entspricht. **.ABC identifiziert beispielsweise alle Objekte mit dem endgültigen Qualifikationsmerkmal ABC.

Profilprioritäten

Ein wichtiger Punkt, der bei der Verwendung generischer Profile zu verstehen ist, ist die Priorität, die bei der Entscheidung, welche Berechtigungen für ein zu erstellendes Objekt angewendet werden sollen, angegeben wird. Angenommen, Sie haben die folgenden Befehle ausgegeben:

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

Der erste Befehl erteilt die Einreihungsberechtigung für alle Warteschlangen für den Principal FRED mit Namen, die dem Profil AB. * entsprechen. Der zweite Befehl erteilt die Abrufberechtigung für dieselben Warteschlangentypen, die dem Profil AB.C*.

Angenommen, Sie erstellen jetzt eine Warteschlange mit dem Namen AB.CD. Entsprechend den Regeln für die Suche nach Platzhalterzeichen kann GRTMQMAUT auf diese Warteschlange angewendet werden. Hat sie also die Befugnis erhalten oder erhalten?

Um die Antwort zu finden, wenden Sie die Regel an, die jedes Mal, wenn mehrere Profile auf ein Objekt angewendet werden können, **nur die spezifischsten gilt** . Die Art und Weise, wie Sie diese Regel anwenden, ist, indem die Profilnamen von links nach rechts verglichen werden. Unabhängig davon, wo sie sich

unterscheiden, ist ein nicht generisches Zeichen spezifischer als ein generisches Zeichen. Im vorherigen Beispiel hat die Warteschlange AB.CD die Berechtigung **get** (AB.C* ist spezifischer als AB. *).

Wenn Sie generische Zeichen vergleichen, lautet die Reihenfolge der *Spezifität* :

1. ?
2. *
3. **

IBM i Installierten Berechtigungsservice unter IBM i angeben

Sie können angeben, welche Berechtigungsservicekomponente verwendet werden soll.

Der Parameter **Service Component name** in **GRTMQMAUT** und **RVKMQMAUT** ermöglicht es Ihnen, den Namen der installierten Berechtigungsservicekomponente anzugeben.

Wenn Sie in der Eingangsanzeige **F24** auswählen, gefolgt von **F9 = Alle Parameter** in der nächsten Anzeige des Befehls, können Sie entweder die installierte Berechtigungskomponente (*DFT) oder den Namen der erforderlichen Berechtigungsservicekomponente angeben, die in der Zeilengruppe 'Service' der Datei 'qm.ini' des WS-Managers angegeben ist.

DSPMQMAUT hat auch diesen zusätzlichen Parameter. Mit diesem Parameter können Sie alle installierten Berechtigungskomponenten (*DFT) oder den angegebenen Berechtigungsservicekomponentennamen für den angegebenen Objektnamen, den Objekttyp und den Benutzer durchsuchen.

IBM i Mit und ohne Berechtigungsprofile unter IBM i arbeiten

In diesem Artikel wird beschrieben, wie mit Berechtigungsprofilen gearbeitet wird und wie ohne Berechtigungsprofile gearbeitet wird.

Wie im Abschnitt „Mit Berechtigungsprofilen arbeiten” auf Seite 195 beschrieben, können Sie mit Berechtigungsprofilen arbeiten, aber auch ohne, wie nachfolgend beschrieben:

Wenn Sie ohne Berechtigungsprofile arbeiten möchten, verwenden Sie *NONE als Berechtigungsparameter unter **GRTMQMAUT** , um Profile ohne Berechtigung zu erstellen. Dadurch bleiben alle vorhandenen Profile unverändert.

Unter **RVKMQMAUT**verwenden Sie *REMOVE als Berechtigungsparameter, um ein vorhandenes Berechtigungsprofil zu entfernen.

Mit Berechtigungsprofilen arbeiten

Es gibt zwei Befehle, die der Berechtigungsprofilierung zugeordnet sind

- **WRKMQMAUT**
- **WRKMQMAUTD**

Sie können diese Befehle direkt über die Befehlszeile oder über die Anzeige WRKMQM aufrufen, indem Sie folgende Schritte ausführen:

1. Geben Sie den Namen des Warteschlangenmanagers ein und drücken Sie die Enter -Taste, um auf die **WRKMQM** -Ergebnisanzeige zuzugreifen.
2. Wählen Sie F23=More options in dieser Anzeige aus.

Option 24 wählt die Ergebnisanzeige für den **WRKMQMAUT** Befehl und Option 25 den Befehl **WRKMQMAUTI** aus, der mit der SSL-Bindungsschicht verwendet wird.

WRKMQMAUT

Mit diesem Befehl können Sie mit den Berechtigungsdaten arbeiten, die in der Berechtigungswarteschlange gespeichert sind.

Anmerkung: Um diesen Befehl ausführen zu können, müssen Sie die Berechtigung `*connect` und `*admdsp` für den Warteschlangenmanager haben. Wenn Sie jedoch ein Profil erstellen oder löschen möchten, benötigen Sie die Berechtigung QMQMADM.

Wenn Sie die Informationen in der Anzeige ausgeben, wird eine Liste der Berechtigungsprofilnamen zusammen mit den zugehörigen Typen angezeigt. Wenn Sie die Ausgabe drucken, erhalten Sie eine detaillierte Liste mit allen Berechtigungsdaten, den registrierten Benutzern und ihren Berechtigungen.

Wenn Sie in dieser Anzeige einen Objekt-oder Profilnamen eingeben und die Eingabetaste drücken, gelangen Sie zur Ergebnisanzeige für **WRKMQMAUT** .

Wenn Sie `4=Delete` auswählen, wechseln Sie in eine neue Anzeige, in der Sie bestätigen können, dass Sie alle Benutzernamen löschen möchten, die in dem von Ihnen angegebenen generischen Berechtigungsprofilnamen registriert sind. Diese Option führt **RVKMQMAUT** mit der Option *REMOVE für alle Benutzer aus und gilt **nur** für generische Profilnamen.

Wenn Sie `12=Work with profile` auswählen, wird die Ergebnisanzeige des Befehls **WRKMQMAUTD** angezeigt. Diese wird im Abschnitt „WRKMQMAUTD” auf Seite 196 näher beschrieben.

WRKMQMAUTD

Mit diesem Befehl können alle Benutzer angezeigt werden, die mit einem bestimmten Berechtigungsprofilnamen und einem bestimmten Objekttyp registriert sind. Um diesen Befehl ausführen zu können, müssen Sie die Berechtigung `*connect` und `*admdsp` für den Warteschlangenmanager haben. Um ein Profil zu erteilen, auszuführen, zu erstellen oder zu löschen, benötigen Sie jedoch die Berechtigung QMQMADM.

Wenn Sie in der Eingangseingabeanzeige `F24=More keys` gefolgt von der Option `F9=All Parameters` auswählen, wird der Name der Servicekomponente wie für **GRTMQMAUT** und **RVKMQMAUT**angezeigt.

Anmerkung: Der `F11=Display Object Authorizations` -Schlüssel schaltet zwischen den folgenden Typen von Berechtigungen um:

- Objektberechtigungen
- Kontextberechtigungen
- MQI-Berechtigungen

Die Optionen in der Anzeige lauten wie folgt:

2=Grant
Ruft die Anzeige **GRTMQMAUT** auf, um die aktuellen Berechtigungen hinzuzufügen.

3=Revoke
Ruft die Anzeige **RVKMQMAUT** auf, um einige der aktuellen Definitionen zu entfernen.

4=Delete
Führt Sie zu einer Anzeige, in der Sie die Berechtigungsdaten für die angegebenen Benutzer löschen können. Dadurch wird **RVKMQMAUT** mit der Option *REMOVE ausgeführt.

5=Display
Führt Sie zum vorhandenen **DSPMQMAUT** -Befehl

F6=Create
Ruft die Anzeige **GRTMQMAUT** auf, in der Sie einen Profilberechtigungssatz erstellen können.

IBM i Richtlinien für den Objektberechtigungsmanager unter IBM i

Zusätzliche Hinweise und Tipps für die Verwendung des Objektberechtigungsmanagers (OAM)

Zugriff auf sensible Operationen begrenzen

Einige Operationen sind sensibel; begrenzen sie auf privilegierte Benutzer. Beispiel:

- Zugriff auf einige spezielle Warteschlangen, wie Übertragungswarteschlangen oder die Befehlswarteschlange `SYSTEM.ADMIN.COMMAND.QUEUE`

- Programme ausführen, die vollständige MQI-Kontextoptionen verwenden
- Anwendungswarteschlangen erstellen und kopieren

WS-Manager-Verzeichnisse

Die Verzeichnisse und Bibliotheken, die Warteschlangen und andere WS-Manager-Daten enthalten, sind privat für das Produkt. Verwenden Sie keine Standardbetriebssystembefehle, um Berechtigungen für MQI-Ressourcen zu erteilen oder zu entziehen.

Warteschlangen

Die Berechtigung für eine dynamische Warteschlange basiert auf, ist aber nicht unbedingt mit der der Modellwarteschlange identisch, aus der sie abgeleitet wurde.

Für Aliaswarteschlangen und ferne Warteschlangen ist die Berechtigung die Berechtigung des Objekts selbst, nicht die Warteschlange, in die der Aliasname oder die ferne Warteschlange aufgelöst wird. Es ist möglich, einem Benutzerprofil die Berechtigung für den Zugriff auf eine Aliaswarteschlange zu erteilen, die in eine lokale Warteschlange aufgelöst wird, für die das Benutzerprofil keine Zugriffsberechtigungen hat.

Begrenzen Sie die Berechtigung zum Erstellen von Warteschlangen für privilegierte Benutzer. Wenn Sie dies nicht tun, können Benutzer die normale Zugriffssteuerung umgehen, indem Sie einen Aliasnamen erstellen.

Alternative Benutzerberechtigung

Die alternative Benutzerberechtigung steuert, ob ein Benutzerprofil die Berechtigung eines anderen Benutzerprofils beim Zugriff auf ein IBM MQ-Objekt verwenden kann. Diese Technik ist wichtig, wenn ein Server Anforderungen von einem Programm empfängt und der Server sicherstellen will, dass das Programm über die erforderliche Berechtigung für die Anforderung verfügt. Der Server verfügt möglicherweise über die erforderliche Berechtigung, aber er muss wissen, ob das Programm über die Berechtigung für die von ihm angeforderten Aktionen verfügt.

For example:

- Ein Serverprogramm, das unter dem Benutzerprofil PAYSERV ausgeführt wird, ruft eine Anforderungsnachricht aus einer Warteschlange ab, die vom Benutzerprofil USER1 in die Warteschlange gestellt wurde.
- Wenn das Serverprogramm die Anforderungsnachricht abruft, verarbeitet es die Anforderung und versetzt die Antwort zurück in die Warteschlange für Antwortnachrichten, die mit der Anforderungsnachricht angegeben ist.
- Anstatt ein eigenes Benutzerprofil (PAYSERV) zu verwenden, um das Öffnen der Warteschlange für Antwortantworten zu autorisieren, kann der Server ein anderes Benutzerprofil, in diesem Fall USER1, angeben. In diesem Beispiel können Sie mit einer alternativen Benutzerberechtigung steuern, ob PAYSERV als alternatives Benutzerprofil USER1 angeben darf, wenn es die Warteschlange für die Antwortwarteschlange öffnet.

Das alternative Benutzerprofil wird im Feld *AlternateUserId* des Objektdeskriptors angegeben.

Anmerkung: Sie können alternative Benutzerprofile für jedes IBM MQ-Objekt verwenden. Die Verwendung eines alternativen Benutzerprofils wirkt sich nicht auf das Benutzerprofil aus, das von einem anderen Ressourcenmanager verwendet wird.

Kontextberechtigung

Kontext ist Informationen, die für eine bestimmte Nachricht gelten und in dem Nachrichtendeskriptor (MQMD) enthalten sind, der Teil der Nachricht ist.

Beschreibungen der Nachrichtendeskriptorfelder, die sich auf den Kontext beziehen, finden Sie in MQMD-Übersicht .

Informationen zu den Kontextoptionen finden Sie im Abschnitt Nachrichtenkontext .

Hinweise zur fernen Sicherheit

Für die ferne Sicherheit ist Folgendes zu beachten:

PUT-Berechtigung
Für die Sicherheit in Warteschlangenmanagern können Sie die Berechtigung "put" angeben, die verwendet wird, wenn ein Kanal eine Nachricht empfängt, die von einem anderen WS-Manager gesendet wird.

Dieser Parameter ist nur für RCVR-, RQSTR-oder CLUSRCVR-Kanaltypen gültig. Geben Sie das Kanalattribut PUTAUT wie folgt an:

DEF
Standardbenutzerprofil. Hierbei handelt es sich um das Benutzerprofil QMQM, unter dem der Nachrichtenkanalagent ausgeführt wird.

CTX
Das Benutzerprofil im Nachrichtenkontext.

Übertragungswarteschlangen
WS-Manager stellen Nachrichten über Fernzugriff automatisch in eine Übertragungswarteschlange. Es ist keine Sonderberechtigung erforderlich. Wenn Sie jedoch eine Nachricht direkt in eine Übertragungswarteschlange stellen, ist eine spezielle Berechtigung erforderlich.

Kanalexits
Kanalexits können für hinzugefügte Sicherheit verwendet werden.

Kanalauthentifizierungsdatensätze
Verwenden Sie diese Option, um eine präzisere Steuerung des Zugriffs zu steuern, der für die Verbindung von Systemen auf Kanalebene erteilt wird.

Weitere Informationen zur fernen Sicherheit finden Sie im Abschnitt „Kanalberechtigung” auf Seite 122.

Kanäle mit SSL/TLS schützen

Das TLS-Protokoll (TLS-Transport Layer Security) bietet Kanalsicherheit mit Schutz vor Ausspionieren, Manipulation und Nachahmungen. Mit der IBM MQ-Unterstützung für TLS können Sie in der Kanaldefinition angeben, dass ein bestimmter Kanal die TLS-Sicherheit verwendet. Sie können auch Details zu der gewünschten Sicherheit angeben, z. B. den Verschlüsselungsalgorithmus, den Sie verwenden möchten.

Die TLS-Unterstützung in IBM MQ verwendet den Warteschlangenmanager *Authentifizierungsdatenobjekt* und verschiedene CL- und MQSC-Befehle sowie Warteschlangenmanager- und Kanalparameter, mit denen die erforderliche TLS-Unterstützung genau definiert wird.

Mit den folgenden CL-Befehlen wird TLS unterstützt:

WRKMQMAUTI
Mit den Attributen eines Authentifizierungsinformationsobjekts arbeiten.

CHGMQMAUTI
Ändern Sie die Attribute eines Authentifizierungsinformationsobjekts.

CRTMQMAUTI
Erstellen Sie ein Authentifizierungsinformationsobjekt.

CPYMQMAUTI
Erstellen Sie ein Authentifizierungsinformationsobjekt, indem Sie ein vorhandenes Objekt kopieren.

DLTMQMAUTI
Authentifizierungsinformationsobjekt löschen.

DSPMQMAUTI
Zeigt die Attribute für ein bestimmtes Authentifizierungsinformationsobjekt an.

Eine Übersicht über die Kanalsicherheit mit TLS finden Sie unter.

- Kanäle mit TLS schützen

Ausführliche Informationen zu den PCF-Befehlen, die TLS zugeordnet sind, finden Sie in.

- Authentifizierungsdatenobjekt ändern, kopieren und erstellen
- Authentifizierungsdatenobjekt löschen
- Authentifizierungsdatenobjekt abfragen

z/OS Sicherheit unter z/OS einrichten

Informationen zu speziellen Sicherheitsaspekten für z/OS.

Die Sicherheit in IBM MQ for z/OS wird mithilfe von RACF oder einem entsprechenden externen Sicherheitsmanager (ESM) gesteuert.

In den folgenden Anweisungen wird vorausgesetzt, dass Sie RACF verwenden.

Zugehörige Verweise

Sicherheitsszenario: zwei Warteschlangenmanager unter z/OS

Sicherheitsszenario: Gruppe mit gemeinsamer Warteschlange unter z/OS

z/OS RACF-Sicherheitsklassen

In RACF-Klassen werden die Profile gespeichert, die für die IBM MQ-Sicherheitsprüfung erforderlich sind. Viele der Mitgliedsklassen verfügen über entsprechende Gruppenklassen. Sie müssen die Klassen aktivieren und sie in die Lage versetzen, generische Profile zu akzeptieren.

Jede RACF-Klasse enthält mindestens ein Profil, das an einem Punkt in der Prüfsequenz verwendet wird, wie in Tabelle 23 auf Seite 199 gezeigt.

Tabelle 23. RACF -Klassen, die von IBM MQ verwendet werden

Mitgliedsklasse	Gruppenklasse	Inhalt
MQADMIN	GMQADMIN	Profile, die hauptsächlich für Verwaltungsfunktionen verwendet werden. Beispiel: • Profile für IBM MQ-Sicherheitsschalter. • Das RESLEVEL-Sicherheitsprofil. • Profile für die Sicherheit alternativer Benutzer-IDs. • Profile für die Kontextsicherheit. • Profile für die Sicherheit von Befehlsressourcen. Diese Klasse kann nur RACF-Profile in Großbuchstaben enthalten.
MXADMIN	GMXADMIN	Profile, die hauptsächlich für Verwaltungsfunktionen verwendet werden. Beispiel: • Profile für IBM MQ-Sicherheitsschalter. • Das RESLEVEL-Sicherheitsprofil. • Profile für die Sicherheit alternativer Benutzer-IDs. • Profile für die Kontextsicherheit. • Profile für die Sicherheit von Befehlsressourcen. Diese Klasse kann RACF-Profile sowohl in Groß- als auch Kleinbuchstaben enthalten.
MQCONN		Profile für die Verbindungssicherheit.

Tabelle 23. RACF -Klassen, die von IBM MQ verwendet werden (Forts.)		
Mitgliedsklasse	**Gruppenklasse**	**Inhalt**
MQCMDS		Profile für die Befehlssicherheit.
MQQUEUE	GMQQUEUE	Profile in Großbuchstaben für die Sicherheit von Warteschlangenressourcen.
MXQUEUE	GMXQUEUE	Profile in gemischter Groß-/Kleinschreibung und in Großbuchstaben für die Sicherheit von Warteschlangenressourcen.
MQPROC	GMQPROC	Profile in Großbuchstaben für die Sicherheit von Prozessressourcen.
MXPROC	GMXPROC	Profile in gemischter Groß- /Kleinschreibung und in Großbuchstaben für die Sicherheit von Prozessressourcen.
MQNLIST	GMQNLIST	Profile in Großbuchstaben für die Ressourcensicherheit von Namenslisten.
MXNLIST	GMXNLIST	Profile in gemischter Groß- /Kleinschreibung und in Großbuchstaben für die Ressourcensicherheit von Namenslisten.
MXTOPIC	GMXTOPIC	Profile in gemischter Groß- /Kleinschreibung und in Großbuchstaben für die Themensicherheit.

Einige Klassen verfügen über eine zugehörige *Gruppenklasse* , mit der Sie Gruppen von Ressourcen zusammenstellen können, die ähnliche Zugriffsvoraussetzungen haben. Details zum Unterschied zwischen den Member- und Gruppenklassen und zum Verwenden einer Member- oder Gruppenklasse finden Sie im Artikel z/OS Security Server RACF Security Administrator's Guide.

Die Klassen müssen aktiviert werden, bevor Sicherheitsprüfungen durchgeführt werden können. Zur Aktivierung aller IBM MQ-Klassen können Sie diesen RACF-Befehl verwenden:

```
SETROPTS CLASSACT(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                  MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

Sie sollten auch sicherstellen, dass Sie die Klassen so konfigurieren, dass sie generische Profile akzeptieren können. Dies erfolgt ebenfalls mithilfe des RACF-Befehls **SETROPTS**, wie zum Beispiel:

```
SETROPTS GENERIC(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                 MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

z/OS RACF-Profile

Alle RACF-Profile, die von IBM MQ verwendet werden, enthalten ein Präfix, bei dem es sich um den Warteschlangenmanagernamen oder den Namen der Gruppe mit gemeinsamer Warteschlange handelt. Gehen Sie vorsichtig vor, wenn Sie das Prozentzeichen als Platzhalterzeichen verwenden.

Alle RACF-Profile, die von IBM MQ verwende werden, enthalten ein Präfix. Bei der Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange ist dies der Name der Gruppe mit gemeinsamer Warteschlange. Für die Sicherheit auf WS-Managerebene ist das Präfix der Name des Warteschlangenmanagers. Wenn Sie eine Kombination der Sicherheit auf Ebene des Warteschlangenmanagers und der Gruppe mit gemeinsamer Warteschlange verwenden, verwenden Sie Profile mit einer Kombination der beiden Präfixtypen. Die Sicherheit für die Gruppe mit gemeinsamer Warteschlange und die Sicherheit auf Warteschlangenmanagerebene werden in Sicherheitssteuerelemente und -optionen in IBM MQ for z/OSbeschrieben.

Wenn Sie beispielsweise eine Warteschlange mit der Bezeichnung QUEUE_FOR_SUBSCRIBER_LIST in der Gruppe QSG1 mit gemeinsamer Warteschlange auf Ebene der Gruppe mit gemeinsamer Warteschlange verwenden möchten, wird das zugehörige Profil für RACF folgendermaßen definiert:

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

Wenn Sie eine Warteschlange mit dem Namen QUEUE_FOR_LOST_CARD_LIST schützen wollen, die Warteschlangenmanager STCD auf Warteschlangenmanagerebene zugeordnet ist, wird das entsprechende Profil für RACF wie folgt definiert:

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

Das bedeutet, dass verschiedene Warteschlangenmanager und Gruppen mit gemeinsamer Warteschlange die gleiche RACF-Datenbank gemeinsam nutzen können und trotzdem verschiedene Sicherheitsoptionen haben.

Verwenden Sie keine generischen WS-Manager-Namen in Profilen, um unerwarteten Benutzerzugriff zu vermeiden.

IBM MQ ermöglicht die Verwendung des Prozentzeichens (%) in Objektnamen. Allerdings verwendet RACF das Prozentzeichen als Einzelzeichenplatzhalter. Dies bedeutet, dass, wenn Sie einen Objektnamen mit einem Prozentzeichen in seinem Namen definieren, dies bei der Definition des entsprechenden Profils berücksichtigt werden müssen.

Für die Warteschlange CREDIT_CARD_%_RATE_INQUIRY auf Warteschlangenmanager CRDP wird das Profil beispielsweise wie folgt für RACF definiert:

```
RDEFINE MQQUEUE CRDP.CREDIT_CARD_%_RATE_INQUIRY
```

Diese Warteschlange kann nicht durch ein generisches Profil geschützt werden, z. B. CRDP. * *.

IBM MQ ermöglicht die Verwendung von gemischter Groß-/Kleinschreibung in Objektnamen. Sie können diese Objekte schützen, indem Sie Folgendes definieren:

1. Profile in gemischter Groß-/Kleinschreibung in den entsprechenden RACF-Klassen mit gemischter Groß-/Kleinschreibung oder
2. Generische Profile in den entsprechenden RACF-Klassen in Großschreibung.

Zur Verwendung von Profilen in gemischter Groß-/Kleinschreibung für RACF-Klassen in gemischter Groß-/Kleinschreibung müssen Sie die in „Einen z/OS -Warteschlangenmanager zur Sicherheit mit gemischter Groß-/Kleinschreibung migrieren” auf Seite 288 beschriebenen Schritte ausführen.

Es gibt einige Profile oder Teile von Profilen, die nur in Großschreibung angegeben werden, da sie Werte von IBM MQ bereitgestellt werden. Diese sind:

- Schalterprofile.
- Alle übergeordneten Qualifikationsmerkmale (HLQ), einschließlich IDs für Subsysteme und Gruppen mit gemeinsamer Warteschlange.
- Profile für SYSTEM-Objekte.
- Profile für Standardobjekte.
- Die Klasse **MQCMDS** , sodass alle Befehlsprofile nur in Großbuchstaben geschrieben werden.
- Die Klasse **MQCONN** , sodass alle Verbindungsprofile nur in Großbuchstaben geschrieben werden.
- **RESLEVEL** -Profile.
- Die Qualifikation von `'object'` in Befehlsressourcenprofilen, z. B. `hlq.QUEUE.queuename`. Der Ressourcenname ist nur in Groß-/Kleinschreibung angegeben.
- Dynamische Warteschlangenprofile `hlq.CSQOREXX.*`, `hlq.CSQUTIL.*` und `CSQXCMD.*`.

- Der Teil `'CONTEXT'` von `hlq.CONTEXT.resourcename`.
- Der Teil `'ALTERNATE.USER'` von `hlq.ALTERNATE.USER.userid`.

Sie können beispielsweise ein Profil definieren, um auf einer der folgenden Arten den Zugriff auf eine Warteschlange mit dem Namen `PAYROLL.Dept1` im Warteschlangenmanager `QM01` zu gewähren.

- Wenn Sie Profile mit gemischter Groß-/Kleinschreibung verwenden, können Sie mit dem folgenden Befehl ein Profil in der IBM MQ RACF-Klasse `MXQUEUE` definieren:

```
RDEFINE MXQUEUE MQ01.PAYROLL.Dept1
```

- Wenn Sie Profile in Großbuchstaben verwenden, können Sie mit dem folgenden Befehl ein Profil in der IBM MQ RACF-Klasse `MQQUEUE` definieren:

```
RDEFINE MQQUEUE MQ01.PAYROLL.*
```

Das erste Beispiel, in dem Profile mit gemischter Groß-/Kleinschreibung verwendet werden, gibt Ihnen eine differenziertere Kontrolle über die Erteilung von Berechtigungen für den Zugriff auf die Ressource.

z/OS Schalterprofile

Wenn Sie die von IBM MQ ausgeführte Sicherheit steuern möchten, verwenden Sie *Schalterprofile*. Ein Schalterprofil ist ein übliches RACF-Profil, das für IBM MQ eine besondere Bedeutung hat. Die Zugriffsliste in Schalterprofilen wird von IBM MQ nicht verwendet.

IBM MQ verwaltet einen internen Schalter für jeden Schaltertyp, der in den Tabellen Profile für Sicherheit auf Subsystemebene umschalten, Profile für die Sicherheit von Gruppen mit gemeinsamer Warteschlange oder Queue-Manager-Ebene-Sicherheit umschalten und Profile für Ressourcenprüfung umschalten gezeigt wird. Schalterprofile können auf Ebene der Gruppe mit gemeinsamer Warteschlange, auf Warteschlangenmanagerebene oder in einer Kombination aus beiden verwaltet werden. Mithilfe einer einzigen Gruppe von Schalterprofilen für die Sicherheit der Gruppe mit gemeinsamer Warteschlange können Sie die Sicherheit auf allen Warteschlangenmanagern innerhalb einer Gruppe mit gemeinsamer Warteschlange steuern.

Wenn ein Sicherheitsschalter festgelegt ist, werden die Sicherheitsprüfungen, die dem Switch zugeordnet sind, ausgeführt. Wenn ein Sicherheitsschalter inaktiviert ist, werden die Sicherheitsprüfungen, die dem Switch zugeordnet sind, umgangen. Der Standardwert ist, dass alle Sicherheitsschalter festgelegt sind.

z/OS *Switches und Klassen*

Beim Start eines Warteschlangenmanagers oder beim Aktualisieren der Sicherheit legt IBM MQ Switches gemäß dem Status der verschiedenen RACF-Klassen fest.

Wenn ein Warteschlangenmanager gestartet wird (oder wenn die Klasse MQADMIN oder MXADMIN mit dem Befehl IBM MQ REFRESH SECURITY aktualisiert wird), überprüft IBM MQ zuerst den Status von RACF und der entsprechenden Klasse:

- Die MQADMIN-Klasse, wenn Sie Profile in Großbuchstaben verwenden
- Die Klasse MXADMIN, wenn Sie ein Profil mit gemischter Groß-/Kleinschreibung verwenden.

Sie setzt die Subsystemsicherheitsfunktion ab, wenn eine der folgenden Bedingungen zutrifft:

- RACF ist inaktiv oder nicht installiert.
- Die Klasse MQADMIN oder MXADMIN ist nicht definiert (diese Klassen sind für RACF immer definiert, da sie in der Klassendeskriptortabelle (CDT) enthalten sind).
- Die Klasse MQADMIN oder MXADMIN wurde nicht aktiviert.

Wenn RACF zusammen mit der Klasse MQADMIN oder MXADMIN aktiv ist, überprüft IBM MQ die Klasse MQADMIN oder MXADMIN, um zu ermitteln, ob Schalterprofile definiert sind. Zunächst werden die im Abschnitt „Profile zur Steuerung der Subsystemsicherheit” auf Seite 203 beschriebenen Profile geprüft.

Wenn keine Subsystemsicherheit erforderlich ist, inaktiviert IBM MQ den Sicherheitsschalter für das interne Subsystem und führt keine weiteren Prüfungen aus.

Die Profile geben an, ob der zugehörige IBM MQ-Schalter ein- oder ausgeschaltet ist.

- Wenn der Schalter ausgeschaltet ist, ist dieser Typ der Sicherheit inaktiviert.
- Wenn ein IBM MQ-Schalter eingeschaltet ist, überprüft IBM MQ den Status der RACF-Klasse, die dem Sicherheitstyp zugeordnet ist, der dem IBM MQ-Schalter entspricht. Wenn die Klasse nicht installiert oder inaktiv ist, wird der IBM MQ-Schalter ausgeschaltet. Prozesssicherheitsprüfungen werden z. B. nicht ausgeführt, wenn die Klasse MQPROC oder MXPROC nicht aktiviert wurde. Die Klasse, die nicht aktiv ist, ist äquivalent zum Definieren des Profils NO.PROCESS.CHECKS für jeden Warteschlangenmanager und jede Gruppe mit gemeinsamer Warteschlange, die diese RACF-Datenbank verwendet.

z/OS *Funktionsweise von Switches*

Definieren Sie NO.*, um einen Sicherheitsschalter zu inaktivieren. Schalterprofil für sie. Sie können NO.* überschreiben. Profilgruppe auf Ebene der Gruppe mit gemeinsamer Warteschlange durch Definition eines YES.* Profil für einen Warteschlangenmanager.

Zum Ausschalten eines Sicherheitsschalters müssen Sie ein NO.* definieren. Schalterprofil für sie. Das Vorhandensein eines NO.* Profil bedeutet, dass Sicherheitsprüfungen für diesen Ressourcentyp **nicht** ausgeführt werden, es sei denn, Sie haben sich entschieden, die Einstellung auf der Ebene der Gruppe mit gemeinsamer Warteschlange für einen bestimmten Warteschlangenmanager zu überschreiben. Dieser Vorgang wird im Abschnitt „Einstellungen auf Ebene der Gruppe mit gemeinsamer Warteschlange überschreiben” auf Seite 203 beschrieben.

Wenn Ihr Warteschlangenmanager kein Mitglied einer Gruppe mit gemeinsamer Warteschlange ist, müssen Sie keine Profile auf Ebene der Gruppe mit gemeinsamer Warteschlange oder andere Überschreibungsprofile definieren. Sie müssen diese Profile allerdings definieren, wenn der Warteschlangenmanager zu einem späteren Zeitpunkt in eine Gruppe mit gemeinsamer Warteschlange aufgenommen wird.

Jede NO.* Das Schalterprofil, das IBM MQ erkennt, inaktiviert die Überprüfung für diesen Ressourcentyp. Schalterprofile werden beim Start des Warteschlangenmanagers aktiviert. Wenn Sie die Schalterprofile ändern, während ein beteiligter Warteschlangenmanager noch ausgeführt wird, kann IBM MQ die Änderungen durch Ausgabe des IBM MQ-Befehls REFRESH SECURITY erkennen.

Die Schalterprofile müssen immer in der Klasse 'MQADMIN' oder 'MXADMIN' definiert sein. Definieren Sie sie nicht in der Klasse GMQADMIN oder GMXADMIN. Tabellen Schalterprofile für Sicherheit auf Subsystemebene und Switchprofile für Ressourcenprüfung zeigen die gültigen Schalterprofile und den Sicherheitstyp an, den sie steuern.

Einstellungen auf Ebene der Gruppe mit gemeinsamer Warteschlange überschreiben

Sie können Sicherheitseinstellungen auf Ebene der Gruppe mit gemeinsamer Warteschlange für einen bestimmten Warteschlangenmanager überschreiben, der Mitglied dieser Gruppe ist. Wenn Sie Warteschlangenmanagerprüfungen für einen einzelnen Warteschlangenmanager ausführen wollen, die nicht für andere Warteschlangenmanager in der Gruppe ausgeführt werden, verwenden Sie den (qmgr-name.YES.*) Schalterprofile.

Wenn Sie hingegen keine bestimmte Prüfung für einen bestimmten Warteschlangenmanager innerhalb einer Gruppe mit gemeinsamer Warteschlange durchführen möchten, definieren Sie einen (qmgr-name.NO.*) Profil für diesen bestimmten Ressourcentyp auf dem Warteschlangenmanager und definieren Sie kein Profil für die Gruppe mit gemeinsamer Warteschlange. (IBM MQ sucht nur dann nach einem Profil auf Ebene der Gruppe mit gemeinsamer Warteschlange, wenn kein Profil auf Warteschlangenmanagerebene gefunden wird.)

z/OS *Profile zur Steuerung der Subsystemsicherheit*

In IBM MQ wird geprüft, ob Sicherheitsprüfungen für das Subsystem, den Warteschlangenmanager und für die Gruppe mit gemeinsamer Warteschlange erforderlich sind.

Mit der ersten Sicherheitsprüfung von IBM MQ wird ermittelt, ob Sicherheitsprüfungen für das gesamte IBM MQ-Subsystem erforderlich sind. Wenn Sie angeben, dass die Subsystemsicherheit nicht verwendet werden soll, werden keine weiteren Prüfungen durchgeführt.

Die folgenden Schalterprofile werden überprüft, um festzustellen, ob die Subsystemsicherheit erforderlich ist. In Abbildung 14 auf Seite 204 ist die Reihenfolge der Prüfungen veranschaulicht.

Tabelle 24. Schalterprofile für Sicherheit auf Subsystemebene

Name des Switch-Profils	**Typ der Ressource oder Überprüfung, die gesteuert wird**
qmgr-name.NO.SUBSYS.SECURITY	Subsystemsicherheit für diesen WS-Manager
qsg-name.NO.SUBSYS.SECURITY	Subsystemsicherheit für diese Gruppe mit gemeinsamer Warteschlange
qmgr-name.YES.SUBSYS.SECURITY	Überschreibung der Subsystemsicherheit für diesen WS-Manager

Wenn Ihr Warteschlangenmanager kein Mitglieder einer Gruppe mit gemeinsamer Warteschlange ist, überprüft IBM MQ nur das Schalterprofil 'qmgr-name.NO.SUBSYS.SECURITY'.

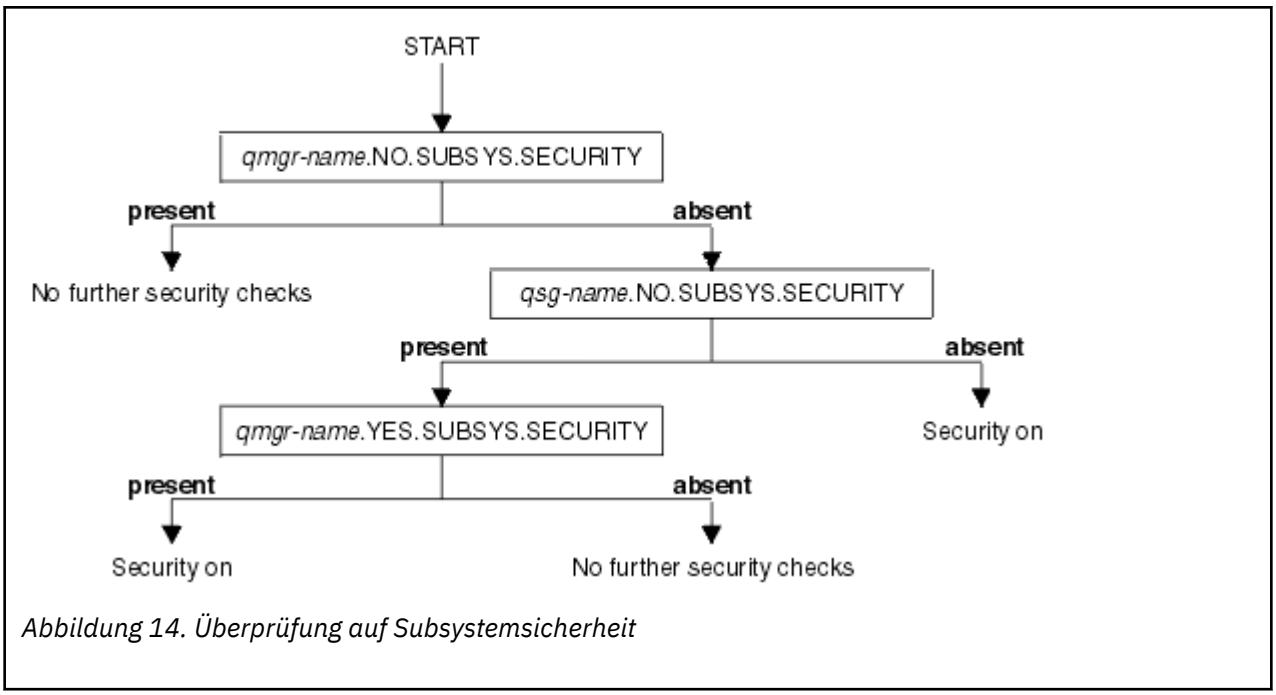


Abbildung 14. Überprüfung auf Subsystemsicherheit

z/OS *Profile zur Steuerung der Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange oder des Warteschlangenmanagers*

Wenn die Sicherheitsprüfung für das Subsystem erforderlich ist, überprüft IBM MQ, ob sie auf Ebene der Gruppe mit gemeinsamer Warteschlange oder auf Warteschlangenmanagerebene ausgeführt werden muss.

Wenn IBM MQ ermittelt hat, dass eine Sicherheitsprüfung erforderlich ist, wird anschließend festgestellt, ob die Prüfung auf Ebene der Gruppe mit gemeinsamer Warteschlange und/oder auf Warteschlangenmanagerebene ausgeführt werden soll. Diese Prüfungen werden nicht vorgenommen, wenn Ihr Warteschlangenmanager kein Mitglied einer Gruppe mit gemeinsamer Warteschlange ist.

Die folgenden Schalterprofile werden überprüft, um die erforderliche Version zu ermitteln. In Abbildung 15 auf Seite 205 und Abbildung 16 auf Seite 206 ist die Reihenfolge der Prüfungen veranschaulicht.

Tabelle 25. Switch-Profil für die Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange oder die Sicherheit auf Warteschlangenmanagerebene

Name des Switch-Profils	**Typ der Ressource oder Überprüfung, die gesteuert wird**
qmgr-name.NO.QMGR.CHECKS	Keine Prüfungen auf WS-Managerebene für diesen WS-Manager
qsg-name.NO.QMGR.CHECKS	Keine Prüfungen auf Warteschlangenmanagerebene für diese Gruppe mit gemeinsamer Warteschlange
qmgr-name.YES.QMGR.CHECKS	Überschreiben der Warteschlangenmanagerebene für diesen WS-Manager
qmgr-name.NO.QSG.CHECKS	Keine Prüfungen auf Ebene der Gruppe mit gemeinsamer Warteschlange für diesen Warteschlangenmanager
qsg-name.NO.QSG.CHECKS	Keine Prüfungen auf Ebene der Gruppe mit gemeinsamer Warteschlange für diese Gruppe mit gemeinsamer Warteschlange
qmgr-name.YES.QSG.CHECKS	Prüfungen auf Ebene der Gruppe mit gemeinsamer Warteschlange werden für diesen Warteschlangenmanager überschrieben

Wenn die Subsystemsicherheit aktiv ist, können Sie die Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange und auf Warteschlangenmanagerebene nicht ausschalten. Falls Sie es dennoch versuchen, aktiviert IBM MQ die Sicherheitsprüfung auf beiden Ebenen.

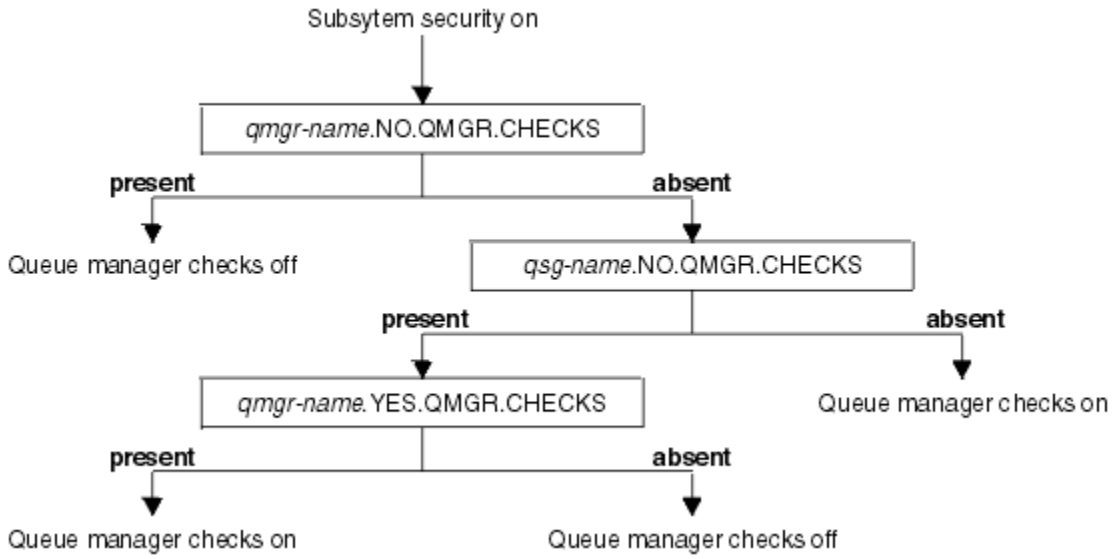


Abbildung 15. Sicherheit auf WS-Managerebene überprüfen

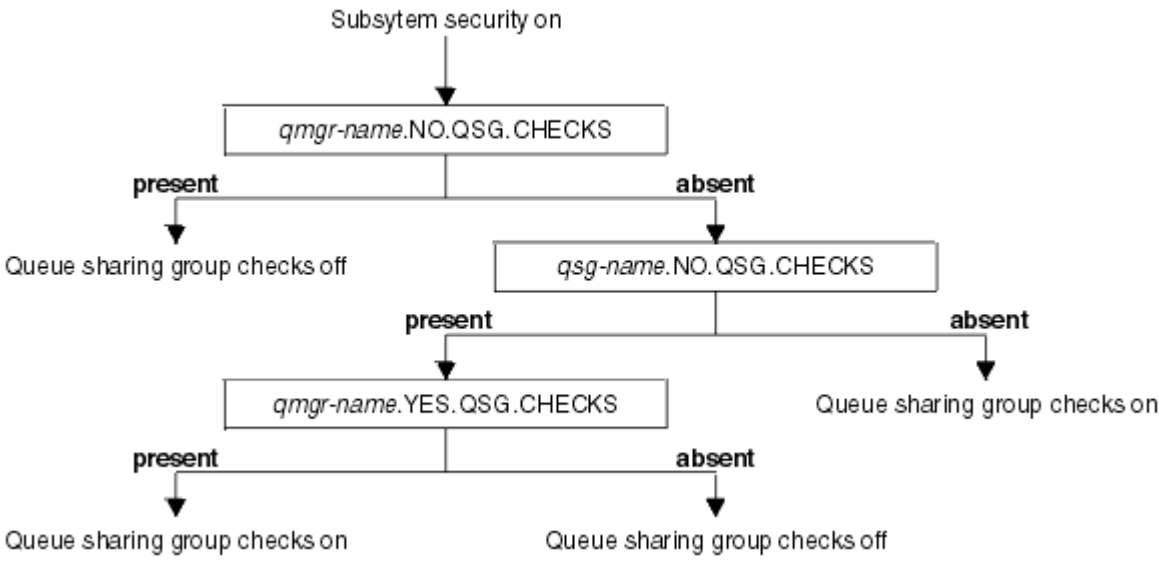


Abbildung 16. Prüfung der Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange

z/OS *Gültige Kombinationen von Sicherheitsschaltern*

Es sind nur bestimmte Kombinationen von Switches gültig. Wenn Sie eine Kombination aus Switch-Einstellungen verwenden, die nicht gültig sind, wird die Nachricht CSQH026I ausgegeben, und die Sicherheitsprüfung wird sowohl auf der Ebene der Gruppe mit gemeinsamer Warteschlange als auch auf Warteschlangenmanagerebene festgelegt.

In Tabelle 26 auf Seite 206, Tabelle 27 auf Seite 206, Tabelle 28 auf Seite 207, und Tabelle 29 auf Seite 207 sind die gültigen Kombinationen von Schaltereinstellungen für jede Art von Sicherheitsstufe aufgeführt.

Tabelle 26. Gültige Sicherheitsschaltkombinationen für die Sicherheit auf WS-Managerebene

Kombinationen
qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS
qsg-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS

Tabelle 27. Gültige Sicherheitsschalterkombinationen für die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange

Kombinationen
qmgr-name.NO.QMGR.CHECKS
qsg-name.NO.QMGR.CHECKS

Tabelle 27. Gültige Sicherheitsschalterkombinationen für die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange (Forts.)
Kombinationen
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS

Tabelle 28. Gültige Sicherheitsschalterkombinationen für die Sicherheit auf Warteschlangenmanagerebene und auf Ebene der Gruppe mit gemeinsamer Warteschlange
Kombinationen
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS Keine QSG.* -Profile definiert
Keine Profile QMGR.* definiert qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
Es sind keine Profile für einen der beiden Schalter definiert

*Tabelle 29. Andere gültige Sicherheitsschalterkombinationen, die beide Ebenen der Überprüfung aktivieren **on** .*
Kombinationen
qmgr-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS

z/OS *Prüfungen auf Ressourcenebene*

Es wird eine Reihe von Schalterprofilen verwendet, um den Zugriff auf Ressourcen zu steuern. Für einige wird die Prüfung in einem Warteschlangenmanager oder in einer Gruppe mit gemeinsamer Warteschlange

gestoppt. Diese können durch Profile überschrieben werden, die die Überprüfung für bestimmte Warteschlangenmanager ermöglichen.

Tabelle 30 auf Seite 208 zeigt die Schalterprofile, mit denen der Zugriff auf IBM MQ-Ressourcen gesteuert wird.

Wenn Ihr Warteschlangenmanager Teil einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit des Warteschlangenmanagers als auch die Sicherheit der Gruppe mit gemeinsamer Warteschlange aktiviert haben, können Sie ein YES.* Wechseln Sie das Profil, um Profile auf Ebene der Gruppe mit gemeinsamer Warteschlange zu überschreiben, und aktivieren Sie speziell die Sicherheit für einen bestimmten Warteschlangenmanager.

Einige Profile gelten für Warteschlangenmanager und Gruppen mit gemeinsamer Warteschlange. In diesen wird die Zeichenfolge *hlq* als Präfix verwendet und Sie sollten den Namen Ihrer Gruppe mit gemeinsamer Warteschlange oder Ihres Warteschlangenmanagers ersetzen, wenn dies möglich ist. Die Profilnamen, die von *qmgr-name* als Präfix angegeben werden, überschreiben die Profile des Warteschlangenmanagers. Sie sollten den Namen des Warteschlangenmanagers ersetzen.

Tabelle 30. Switchprofile für Ressourcenüberprüfung

Typ der Ressourcenüberprüfung, die gesteuert wird	**Name des Switch-Profils**	**Profil für einen bestimmten WS-Manager überschreiben**
Verbindungssicherheit	hlq.NO.CONNECT.CHECKS	qmgr-name.YES.CONNECT.CHECKS
Warteschlangensicherheit	hlq.NO.QUEUE.CHECKS	qmgr-name.YES.QUEUE.CHECKS
Prozesssicherheit	hlq.NO.PROCESS.CHECKS	qmgr-name.YES.PROCESS.CHECKS
Namensliste, Sicherheit	hlq.NO.NLIST.CHECKS	qmgr-name.YES.NLIST.CHECKS
Kontextsicherheit	hlq.NO.CONTEXT.CHECKS	qmgr-name.YES.CONTEXT.CHECKS
Alternative Benutzersicherheit	hlq.NO.ALTERNATE.USER.CHECKS	qmgr-name.YES.ALTERNATE.USER.CHECKS
Befehlssicherheit	hlq.NO.CMD.CHECKS	qmgr-name.YES.CMD.CHECKS
Befehlsressourcensicherheit	hlq.NO.CMD.RESC.CHECKS	qmgr-name.YES.CMD.RESC.CHECKS
Themensicherheit	hlq.NO.TOPIC.CHECKS	qmgr-name.YES.TOPIC.CHECKS
Anmerkung: Generische Schalterprofile wie hlq.NO. * * werden von IBM MQ ignoriert		

Wenn Sie beispielsweise Prüfungen der Prozesssicherheit auf dem Warteschlangenmanager QM01 ausführen möchten, der Mitglied der Gruppe QSG3 mit gemeinsamer Warteschlange ist, Sie aber keine Prüfungen der Prozesssicherheit auf den anderen Warteschlangenmanagern in der Gruppe vornehmen möchten, definieren Sie die folgenden Schalterprofile:

```
QSG3.NO.PROCESS.CHECKS
QM01.YES.PROCESS.CHECKS
```

Wenn die Prüfungen der Warteschlangensicherheit auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange mit Ausnahme von QM02 ausgeführt werden sollen, definieren Sie die folgenden Schalterprofile:

```
QM02.NO.QUEUE.CHECKS
```

(Es muss kein Profil für die Gruppe mit gemeinsamer Warteschlange definiert werden, da die Prüfungen automatisch aktiviert sind, wenn dort kein Profil definiert ist.)

z/OS *Beispiel für die Definition von Switches*

Unterschiedliche IBM MQ-Subsysteme haben unterschiedliche Anforderungen, die mit unterschiedlichen Schalterprofilen implementiert werden können.

Es wurden vier IBM MQ-Subsysteme definiert:

- MQP1 (Produktionssystem)
- MQP2 (Produktionssystem)
- MQD1 (Entwicklungssystem)
- MQT1 (Testsystem)

Alle vier Warteschlangenmanager sind Mitglieder der Gruppe mit gemeinsamer Warteschlange QS01. Alle IBM MQ RACF-Klassen wurden definiert und aktiviert.

Diese Subsysteme haben unterschiedliche Sicherheitsanforderungen:

- Für Produktionssysteme muss die vollständige Überprüfung der IBM MQ-Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange auf beiden Systemen aktiv sein.

  Geben Sie dazu das folgende Profil an:

  ```
  RDEFINE MQADMIN QS01.NO.QMGR.CHECKS
  ```

  Dadurch wird die Überprüfung auf Ebene der Gruppe mit gemeinsamer Warteschlange für alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange festgelegt. Sie müssen keine anderen Schalterprofile für die Produktionswarteschlangenmanager definieren, da Sie alles für diese Systeme überprüfen wollen.
- Der Testwarteschlangenmanager MQT1 erfordert auch eine vollständige Sicherheitsprüfung. Da Sie dies später möglicherweise ändern möchten, kann die Sicherheit auf Warteschlangenmanagerebene definiert werden, damit Sie die Sicherheitseinstellungen für diesen Warteschlangenmanager ändern können, ohne die anderen Mitglieder der Gruppe mit gemeinsamer Warteschlange zu beeinträchtigen.

  Dies wird durch die Definition des Profils NO.QSG.CHECKS für MQT1 wie folgt definiert:

  ```
  RDEFINE MQADMIN MQT1.NO.QSG.CHECKS
  ```
- Der Warteschlangenmanager MQD1 für die Entwicklung hat andere Sicherheitsanforderungen als die übrigen Mitglieder der Gruppe mit gemeinsamer Warteschlange. Es ist nur erforderlich, dass die Verbindung und die Warteschlangensicherheit aktiv sind.

  Dazu definieren Sie ein `MQD1.YES.QMGR.CHECKS` -Profil für diesen Warteschlangenmanager und definieren anschließend die folgenden Profile, um die Sicherheitsprüfung für die Ressourcen, die nicht überprüft werden müssen, abzuschalten:

  ```
  RDEFINE MQADMIN MQD1.NO.CMD.CHECKS
  RDEFINE MQADMIN MQD1.NO.CMD.RESC.CHECKS
  RDEFINE MQADMIN MQD1.NO.PROCESS.CHECKS
  RDEFINE MQADMIN MQD1.NO.NLIST.CHECKS
  RDEFINE MQADMIN MQD1.NO.CONTEXT.CHECKS
  RDEFINE MQADMIN MQD1.NO.ALTERNATE.USER.CHECKS
  ```

Wenn der WS-Manager aktiv ist, können Sie die aktuellen Sicherheitseinstellungen anzeigen, indem Sie den Befehl DISPLAY SECURITY MQSC ausgeben.

Sie können die Schaltereinstellungen auch ändern, wenn der Warteschlangenmanager ausgeführt wird, indem Sie das entsprechende Schalterprofil in der Klasse MQADMIN definieren oder löschen. Um die Änderungen an den Schaltereinstellungen aktiv zu machen, müssen Sie den Befehl REFRESH SECURITY für die Klasse MQADMIN absetzen.

Weitere Informationen zur Verwendung der Befehle DISPLAY SECURITY und REFRESH SECURITY finden Sie im Abschnitt „Sicherheit des Warteschlangenmanagers unter z/OS aktualisieren” auf Seite 268.

z/OS Profile, die zum Steuern des Zugriffs auf IBM MQ-Ressourcen verwendet werden

Sie müssen RACF-Profile definieren, um den Zugriff auf IBM MQ-Ressourcen zusätzlich zu den möglicherweise definierten Schalterprofilen zu steuern. Diese Themensammlung enthält Informationen zu den RACF-Profilen für die verschiedenen IBM MQ-Ressourcentypen.

Wenn für eine bestimmte Sicherheitsprüfung kein Ressourcenprofil definiert ist und ein Benutzer eine Anforderung ausgibt, in deren Rahmen die betreffende Prüfung erforderlich wäre, verweigert IBM MQ den Zugriff. Es müssen keine Profile für Sicherheitstypen definiert werden, die sich auf Sicherheitsschalter beziehen, die Sie inaktiviert haben.

z/OS *Profile für Verbindungssicherheit*

Wenn die Verbindungssicherheit aktiv ist, müssen Sie Profile in der Klasse MQCONN definieren und den erforderlichen Gruppen oder Benutzer-IDs den Zugriff auf diese Profile erlauben, damit sie eine Verbindung zu IBM MQ herstellen können.

Damit eine Verbindung hergestellt werden kann, müssen Sie Benutzern RACF-Lesezugriff (READ) für das entsprechende Profil erteilen. (Wenn kein Profil auf Warteschlangenmanagerebene vorhanden ist und Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist, werden möglicherweise Prüfungen für das Profil auf Ebene der Gruppe mit gemeinsamer Warteschlange ausgeführt, wenn die Sicherheit dafür eingerichtet ist.)

Ein Verbindungsprofil, das mit einem Warteschlangenmanager-Namen qualifiziert ist, steuert den Zugriff auf einen bestimmten Warteschlangenmanager, und Benutzer, die Zugriff auf dieses Profil haben, können eine Verbindung zu diesem Warteschlangenmanager herstellen. Ein Verbindungsprofil, das mit dem Namen der Gruppe mit gemeinsamer Warteschlange qualifiziert wurde, steuert den Zugriff auf alle Warteschlangenmanager innerhalb der Gruppe mit gemeinsamer Warteschlange für diesen Verbindungstyp. Beispielsweise kann ein Benutzer mit Zugriff auf `QS01.BATCH` eine Stapelverbindung zu jeden Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange QS01 verwenden, für die kein Profil auf Warteschlangenmanagerebene definiert ist.

Anmerkung:

1. Informationen zu den für verschiedene Sicherheitsanforderungen geprüften Benutzer-IDs finden Sie im Abschnitt „Benutzer-IDs für die Sicherheitsprüfung unter z/OS” auf Seite 255.
2. Die Sicherheitsprüfungen auf Ressourcenebene (RESLEVEL) werden ebenfalls zur Verbindungszeit durchgeführt. Weitere Informationen finden Sie unter „Sicherheitsprofil RESLEVEL” auf Seite 249.

Die IBM MQ-Sicherheit erkennt die folgenden unterschiedlichen Verbindungstypen:

- Stapelverbindungen (und Stapelverbindungen), die Folgendes umfassen:
  - z/OS-Batch-Jobs
  - TSO-Anwendungen
  - z/OS UNIX System Services-Anmeldungen
  - Gespeicherte Db2-Prozeduren
- CICS-Verbindungen
- IMS-Verbindungen aus Steuerungs- und Anwendungsverarbeitungsregionen
- Der IBM MQ-Kanalinitiator

z/OS *Verbindungssicherheitsprofile für Stapelverbindungen*

Profile für die Überprüfung von Stapelverbindungen bestehen aus dem Namen des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange gefolgt vom Wort *BATCH*. Geben Sie der Benutzer-ID, die dem Verbindungsadressraum READ zugeordnet ist, Lesezugriff auf das Verbindungsprofil an.

Profile zum Prüfen von Batch-und Batch-Typ-Verbindungen haben das folgende Format:

```
hlq.BATCH
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein. Wenn Sie die Sicherheit auf Ebene des Warteschlangenmanagers und der Gruppe mit gemeinsamer Warteschlange verwenden, überprüft IBM MQ, ob ein Profil mit dem Warteschlangenmanagernamen als Präfix vorhanden ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist. Wenn das Profil nicht gefunden werden kann, schlägt die Verbindungsanforderung fehl.

Für Stapel-oder Stapelverbindungsanforderungen müssen Sie zulassen, dass die Benutzer-ID, die dem Verbindungsadressbereich zugeordnet ist, auf das Verbindungsprofil zugreifen kann. Mit dem folgenden RACF-Befehl beispielsweise werden Benutzer in der Gruppe CONNTQM1 berechtigt, eine Verbindung zu Warteschlangenmanager TQM1 herzustellen; diese Benutzer-IDs dürfen dann alle Stapelverbindungen oder Verbindungen dieses Typs verwenden.

```
RDEFINE MQCONN TQM1.BATCH UACC(NONE)
PERMIT TQM1.BATCH CLASS(MQCONN) ID(CONNTQM1) ACCESS(READ)
```

z/OS ***CHCKLOCL** für lokal gebundene Anwendungen verwenden*

CHCKLOCL gilt nur für Verbindungen, die über BATCH-Verbindungen vorgenommen wurden, jedoch nicht für Verbindungen aus CICS oder IMS. Verbindungen, die über den Kanalinitiator hergestellt werden, werden von **CHCKCLNT** gesteuert.

Übersicht

Wenn Sie Ihre z/OS-Warteschlangenmanager so konfigurieren möchten, dass nur einige, aber nicht alle Benutzer-IDs und Kennwörter ihrer lokal gebundenen Anwendungen geprüft werden, müssen Sie einige zusätzliche Konfigurationsschritte ausführen.

Der Grund hierfür ist, dass traditionelle Stapelanwendungen, die den MQCONN-API-Aufruf verwenden, nicht mehr mit dem WS-Manager verbunden werden können, wenn **CHCKLOCL** (*REQUIRED*) konfiguriert ist.

Nur für z/OS kann ein differenzierteres Verfahren auf Basis der Verbindungssicherheit eines Adressraums verwendet werden, um die globale Konfiguration CHCKLOCL(REQUIRED) für ausdrücklich definierte Benutzer-IDs auf CHCKLOCL(OPTIONAL) herabzustufen. Der verwendete Mechanismus wird in dem folgenden Text zusammen mit einem Beispiel beschrieben.

Um mehr Granularität auf **CHCKLOCL** (*REQUIRED*) als nur EVERYONE zu ermöglichen, ändern Sie **CHCKLOCL** in der gleichen Weise wie die Zugriffsebene der Benutzer-ID, die dem Verbindungsadressraum zugeordnet ist, zu den `hlq.batch` -Verbindungsprofilen in der Klasse MQCONN.

Wenn die Benutzer-ID des Adressraums nur über Lesezugriff verfügt. Dies ist das Minimum, das Sie benötigen, um überhaupt eine Verbindung herstellen zu können. Die **CHCKLOCL** -Konfiguration gilt wie geschrieben.

Wenn die Benutzer-ID des Adressraums UPDATE-Zugriff (oder höher) hat, dann wird die **CHCKLOCL** -Konfiguration im *OPTIONAL* -Modus ausgeführt. Das heißt, Sie müssen keine Benutzer-ID und kein Kennwort angeben, aber wenn Sie dies tun, müssen die Benutzer-ID und das Kennwort ein gültiges Paar sein.

Verbindungssicherheit ist für den z/OS-Warteschlangenmanager bereits konfiguriert

Wenn die Verbindungssicherheit für Ihren z/OS-Warteschlangenmanager konfiguriert ist und **CHCKLOCL** (*REQUIRED*) nur für lokal gebundene WAS-Anwendungen angewendet werden soll, führen Sie die folgenden Schritte aus:

1. Beginnen Sie mit **CHCKLOCL** (*OPTIONAL*) als Ihre Konfiguration. Dies bedeutet, dass jede Benutzer-ID und Kennwörter, die zur Verfügung gestellt werden, auf Gültigkeit geprüft, aber nicht in der vorgeschriebenen Weise angegeben werden.
2. Listen Sie alle Benutzer mit Zugriff auf die Verbindungssicherheitsprofile auf, indem Sie den folgenden Befehl ausgeben:

   ```
   RLIST MQCONN MQ23.BATCH AUTHUSER
   ```

   Dieser Befehl wird angezeigt, z. B.:

   ```
   CLASS    NAME
   -----    ----
   MQCONN   MQ23.BATCH

   USER     ACCESS  ACCESS COUNT
   ----     ------  ------ -----
   JOHNDOE  READ    000009
   JDOE1    READ    000003
   WASUSER  READ    000000
   ```

3. Ändern Sie für jede Benutzer-ID, die als READ-Zugriff aufgeführt ist, den Zugriff auf

   ```
   UPDATE:- PERMIT MQ23.BATCH CLASS(MQCONN) ID(JOHNDOE) ACCESS(UPDATE)
   ```

4. Aktualisieren Sie die IBM MQ-Konfiguration auf **CHCKLOCL** (*REQUIRED*).

   Die Kombination aus UPDATE-Zugriff auf `MQ23.BATCH` und der aktuellen Einstellung bedeutet, dass Sie **CHCKLOCL** (*OPTIONAL*) verwenden.
5. Nun wenden Sie das Verhalten von **CHCKLOCL** (*REQUIRED*) auf eine bestimmte Benutzer-ID, z. B. WASUSER, an, sodass alle Verbindungen, die aus dieser Region stammen, eine Benutzer-ID und ein Kennwort bereitstellen müssen.

   Führen Sie den folgenden Befehl aus, um die zuvor vorgenommenen Änderungen rückgängig zu machen:

   ```
   PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
   ```

Verbindungssicherheit ist für den z/OS-Warteschlangenmanager nicht konfiguriert

In dieser Situation müssen Sie:

1. Erstellen Sie in der MQCONN-Klasse Verbindungsprofile für `hlq.BATCH` , indem Sie den folgenden Befehl ausgeben:

   ```
   RDEFINE MQCONN MQ23.BATCH UACC(NONE)
   ```

2. Berechtigen Sie alle Benutzer-IDs, die Batchverbindungen zum Warteschlangenmanager erstellen, so dass sie über UPDATE-Zugriff auf dieses Profil verfügen. Dadurch wird die **CHCKLOCL** (*REQUIRED*) -Anforderung für die Benutzer-ID und das Kennwort zum Zeitpunkt der Verbindung umgangen.

   Geben Sie dazu den folgenden Befehl aus:

   ```
   PERMIT MQ23.BATCH CLASS(MQCONN)ID(JOHNDOE) ACCESS(UPDATE)
   ```

   Hierzu gehören die folgenden Benutzer-IDs:

   a. Wird für CSQUTIL, ISPF-Anzeigen und andere lokal gebundene Tools verwendet.
   b. Zugeordnete Stapelverarbeitung wie Verbindungen zum Warteschlangenmanager. Beispiel: Advanced Message Security, IBM Integration Bus, gespeicherte Db2-Prozeduren, z/OS UNIX System Services und TSO-Benutzer und Java-Anwendungen

3. Löschen Sie das Schalterprofil für den Warteschlangenmanager, indem Sie den folgenden Befehl ausgeben:

```
hlq.NO.CONNECT.CHECKS
```

4. Nun wenden Sie das Verhalten von **CHCKLOCL** (*REQUIRED*) auf eine bestimmte Benutzer-ID, z. B. WASUSER, an, sodass alle Verbindungen, die aus dieser Region stammen, eine Benutzer-ID und ein Kennwort bereitstellen müssen.

   Führen Sie den folgenden Befehl aus, um die zuvor vorgenommenen Änderungen rückgängig zu machen:

```
PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
```

z/OS *Verbindungssicherheitsprofile für CICS-Verbindungen*

Profile für die Überprüfung von CICS-Verbindungen bestehen aus dem Namen des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange gefolgt von dem Wort *CICS* . Erteilen Sie der Benutzer-ID, die dem CICS-Adressraum zugeordnet ist, Lesezugriff (READ) auf das Verbindungsprofil.

Profile für die Überprüfung von Verbindungen von CICS haben das folgende Format:

```
hlq.CICS
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein. Wenn Sie die Sicherheit auf Ebene des Warteschlangenmanagers und der Gruppe mit gemeinsamer Warteschlange verwenden, überprüft IBM MQ, ob ein Profil mit dem Warteschlangenmanagernamen als Präfix vorhanden ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist. Wenn das Profil nicht gefunden wird, schlägt die Verbindungsanforderung fehl.

Für Verbindungsanforderungen von CICS müssen Sie nur der Benutzer-ID für den CICS-Adressraum Zugriff auf das Verbindungsprofil erteilen.

Mit den folgenden RACF-Befehlen kann die Benutzer-ID KCBCICS für den CICS-Adressraum beispielsweise eine Verbindung zu Warteschlangenmanager TQM1 herstellen:

```
RDEFINE MQCONN TQM1.CICS UACC(NONE)
PERMIT TQM1.CICS CLASS(MQCONN) ID(KCBCICS) ACCESS(READ)
```

z/OS *Verbindungssicherheitsprofile für IMS-Verbindungen*

Profile für die Überprüfung von IMS-Verbindungen bestehen aus dem Namen des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange gefolgt von dem Wort *IMS* . Erteilen Sie der IMS-Steuerung und den Benutzer-IDs für die abhängige Region Lesezugriff (READ) auf das Verbindungsprofil.

Profile für die Überprüfung von Verbindungen von IMS haben das folgende Format:

```
hlq.IMS
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein. Wenn Sie die Sicherheit auf Ebene des Warteschlangenmanagers und der Gruppe mit gemeinsamer Warteschlange verwenden, überprüft IBM MQ, ob ein Profil mit dem Warteschlangenmanagernamen als Präfix vorhanden ist. Wenn er kein Profil gefunden hat, sucht er

nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist. Wenn das Profil nicht gefunden wird, schlägt die Verbindungsanforderung fehl.

Für Verbindungsanforderungen von IMS können Sie den Zugriff auf das Verbindungsprofil für die IMS-Steuerung und Benutzer-IDs der abhängigen Regionen erteilen.

Mit den folgenden RACF-Befehlen werden beispielsweise folgende Berechtigungen erteilt:

- Die Benutzer-ID der IMS-Region (IMSREG), zum Herstellen einer Verbindung zu Warteschlangenmanager TQM1.
- Benutzer in der Gruppe BMPGRP zum Übergeben von BMP-Jobs.

```
RDEFINE MQCONN TQM1.IMS UACC(NONE)
PERMIT TQM1.IMS CLASS(MQCONN) ID(IMSREG,BMPGRP) ACCESS(READ)
```

z/OS *Verbindungssicherheitsprofile für den Kanalinitiator*

Profile für die Überprüfung von Verbindungen vom Kanalinitiator bestehen aus dem Namen des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange, gefolgt von dem Wort *CHIN*. Geben Sie die Benutzer-ID, die vom Kanalinitiator verwendet wird, den Adressraum READ für den Taskadressbereich READ für das Verbindungsprofil an.

Profile zum Überprüfen von Verbindungen aus dem Kanalinitiator haben das folgende Format:

```
hlq.CHIN
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein. Wenn Sie die Sicherheit auf Ebene des Warteschlangenmanagers und der Gruppe mit gemeinsamer Warteschlange verwenden, überprüft IBM MQ, ob ein Profil mit dem Warteschlangenmanagernamen als Präfix vorhanden ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist. Wenn das Profil nicht gefunden wird, schlägt die Verbindungsanforderung fehl.

Definieren Sie für Verbindungsanforderungen durch den Kanalinitiator den Zugriff auf das Verbindungsprofil für die Benutzer-ID, die von dem Adressraum der Kanalinitiatortask verwendet wird.

Mit den folgenden RACF-Befehlen beispielsweise wird der unter der Benutzer-ID DQCTRL ausgeführte Adressraum des Kanalinitiators berechtigt, eine Verbindung zu Warteschlangenmanager TQM1 herzustellen:

```
RDEFINE MQCONN TQM1.CHIN UACC(NONE)
PERMIT TQM1.CHIN CLASS(MQCONN) ID(DQCTRL) ACCESS(READ)
```

z/OS *Profile für die Warteschlangensicherheit*

Wenn die Warteschlangensicherheit aktiv ist, müssen Sie Profile in den entsprechenden Klassen definieren und die erforderlichen Gruppen oder Benutzer-IDs auf diese Profile zugreifen lassen. Profile für die Warteschlangensicherheit werden nach dem Warteschlangenmanager oder der Gruppe mit gemeinsamer Warteschlange sowie der Warteschlange benannt, die geöffnet werden soll.

Wenn die Warteschlangensicherheit aktiv ist, müssen Sie:

- Definieren Sie Profile in den Klassen **MQQUEUE** oder **GMQQUEUE** , wenn Sie Profile in Großbuchstaben verwenden.
- Definieren Sie Profile in den Klassen **MXQUEUE** oder **GMXQUEUE** , wenn Sie Profile mit Groß-/Kleinschreibung verwenden.

- Erteilen Sie den erforderlichen Gruppen oder Benutzer-IDs Zugriff auf diese Profile, damit sie IBM MQ-API-Anforderungen ausgeben können, die Warteschlangen verwenden.

Profile für die Warteschlangensicherheit haben das folgende Format:

```
hlq.queuename
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein und `queuename` ist der Name der Warteschlange, die geöffnet wird, wie im Objektdeskriptor im Aufruf MQOPEN oder MQPUT1 angegeben.

Ein Profil, dem der Name des Warteschlangenmanagers vorangesetzt ist, steuert den Zugriff auf eine einzelne Warteschlange in diesem Warteschlangenmanager. Ein Profil mit dem Namen der Gruppe mit gemeinsamer Warteschlange als Präfix steuert den Zugriff auf eine oder mehrere Warteschlangen mit diesem Warteschlangennamen auf allen Warteschlangenmanagern innerhalb der Gruppe mit gemeinsamer Warteschlange oder den Zugriff auf eine gemeinsam genutzte Warteschlange durch einen Warteschlangenmanager in der Gruppe. Dieser Zugriff kann auf einem einzelnen WS-Manager außer Kraft gesetzt werden, indem ein WS-Managerebenenprofil für diese Warteschlange in diesem Warteschlangenmanager definiert wird.

Wenn Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit auf der Ebene des Warteschlangenmanagers als auch die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, sucht IBM MQ zuerst nach einem Profil, das dem Namen des Warteschlangenmanagers vorangestellt ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist.

Wenn Sie gemeinsam genutzte Warteschlangen verwenden, wird die Verwendung der Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange empfohlen.

Einzelheiten zur Funktionsweise der Warteschlangensicherheit, wenn der Warteschlangenname der Name einer Alias- oder Modellwarteschlange ist z/OS, finden Sie in den Abschnitten „Hinweise zu Aliaswarteschlangen” auf Seite 217 und „Hinweise zu Modellwarteschlangen” auf Seite 218 .

Der zum Öffnen einer Warteschlange erforderliche RACF-Zugriff hängt von den angegebenen Optionen MQOPEN oder MQPUT1 ab. Wenn mehr als eine der Optionen MQOO_* und MQPMO_* codiert ist, wird die Sicherheitsprüfung für die höchste erforderliche RACF-Berechtigung ausgeführt.

Tabelle 31. Zugriffsebenen für die Warteschlangensicherheit mit den MQOPEN-oder MQPUT1-Aufrufen

MQOPEN-oder MQPUT1-Option	**RACF-Zugriffsebene, die für 'hlq.queuename' erforderlich ist**
MQOO_BROWSE	READ
MQOO_INQUIRE	READ
MQOO_BIND_ *	UPDATE
MQOO_INPUT_*	UPDATE
MQOO_OUTPUT oder MQPUT1	UPDATE
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	UPDATE
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	UPDATE
MQOO_SAVE_ALL_CONTEXT	UPDATE
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE

Tabelle 31. Zugriffsebenen für die Warteschlangensicherheit mit den MQOPEN-oder MQPUT1-Aufrufen (Forts.)

MQOPEN-oder MQPUT1-Option	**RACF-Zugriffsebene, die für 'hlq.queuename' erforderlich ist**
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	UPDATE
MQOO_SET	ALTER

Auf dem IBM MQ-Queue Manager QM77 sollen beispielsweise alle Benutzer-IDs in der RACF-Gruppe PAYGRP Zugriff erhalten, um Nachrichten von allen Warteschlangen mit Namen zu erhalten, die mit 'PAY.' beginnen, oder Nachrichten an alle Warteschlangen zu stellen. Sie können dies mit den folgenden RACF-Befehlen vornehmen:

```
RDEFINE MQQUEUE QM77.PAY.** UACC(NONE)
PERMIT QM77.PAY.** CLASS(MQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Außerdem müssen alle Benutzer-IDs in der PAYGRP-Gruppe Zugriff haben, um Nachrichten in Warteschlangen zu stellen, die nicht der PAY-Namenskonvention entsprechen. For example:

```
REQUEST_QUEUE_FOR_PAYROLL
SALARY.INCREASE.SERVER
REPLIES.FROM.SALARY.MODEL
```

Sie können dies tun, indem Sie Profile für diese Warteschlangen in der Klasse GMQQUEUE definieren und den Zugriff auf diese Klasse wie folgt erteilen:

```
RDEFINE GMQQUEUE PAYROLL.EXTRAS UACC(NONE)
        ADDMEM(QM77.REQUEST_QUEUE_FOR_PAYROLL,
               QM77.SALARY.INCREASE.SERVER,
               QM77.REPLIES.FROM.SALARY.MODEL)
PERMIT PAYROLL.EXTRAS CLASS(GMQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Anmerkung:

1. Wenn die RACF-Zugriffsebene geändert wird, die eine Anwendung für das Sicherheitsprofil einer Warteschlange hat, werden die Änderungen nur für die neuen Objektkennungen wirksam (also neue MQOPEN-Objekte), die für diese Warteschlange angefordert werden. Diese Kennungen, die zum Zeitpunkt der Änderung bereits vorhanden sind, behalten ihren vorhandenen Zugriff auf die Warteschlange bei. Wenn eine Anwendung benötigt wird, um ihre geänderte Zugriffsebene in die Warteschlange anstatt auf die vorhandene Zugriffsebene zu verwenden, muss sie die Warteschlange für jede Objektkennung, die die Änderung erfordert, schließen und erneut öffnen.
2. Im Beispiel könnte der Warteschlangenmanagername QM77 auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Es können auch andere Arten von Sicherheitsprüfungen auftreten, wenn die Warteschlange in Abhängigkeit von den angegebenen Öffnungsoptionen und den aktiven Sicherheitstypen geöffnet wird.

z/OS Weitere Informationen finden Sie unter „Profile für Kontextsicherheit” auf Seite 233 und „Profile für alternative Benutzersicherheit” auf Seite 231. Eine Übersichtstabelle über die Optionen für das Öffnen sowie über die erforderliche Sicherheitsberechtigung, wenn sowohl die Wartschlangensicherheit als auch die Kontextsicherheit und die Sicherheit der alternativen Benutzer-IDs aktiv sind, finden Sie im Abschnitt Tabelle 36 auf Seite 223.

Wenn Sie Publish/Subscribe verwenden, müssen Sie die folgenden Informationen beachten. Bei der Verarbeitung einer MQSUB-Anforderung wird eine Sicherheitsprüfung ausgeführt, um sicherzustellen, dass die Benutzer-ID, von der die Anforderung gestellt wird, berechtigt ist, Nachrichten in die IBM MQ-Zielwarteschlange einzureihen und das IBM MQ-Thema zu subskribieren.

Tabelle 32. Zugriffsebenen für die Warteschlangensicherheit mit dem MQSUB-Aufruf

MQSUB, Option	**RACF-Zugriffsebene, die für 'hlq.queuename' erforderlich ist**
MQSO_ALTER, MQSO_CREATE und MQSO_RESUME	UPDATE

Anmerkung:

1. Der `hlq.queuename` ist die Zielwarteschlange für Veröffentlichungen. Wenn es sich um eine verwaltete Warteschlange handelt, benötigen Sie Zugriff auf die entsprechende Modellwarteschlange, die für die verwaltete Warteschlange und die dynamische Warteschlange, die erstellt werden, verwendet werden soll.
2. Sie können ein Verfahren wie dieses für die Zielwarteschlange verwenden, die Sie in einem MQSUB-API-Aufruf bereitstellen, wenn Sie zwischen den Benutzern, die die Subskriptionen vornehmen, und den Benutzern, die die Veröffentlichungen abrufen, von der Zielwarteschlange unterscheiden wollen.

z/OS *Hinweise zu Aliaswarteschlangen*

Bei der Ausgabe eines `MQOPEN`- oder `MQPUT1`-Aufrufs für eine Aliaswarteschlange führt IBM MQ eine Ressourcenprüfung für den im Objektdeskriptor (MQOD) des Aufrufs angegebenen Warteschlangemanager aus. Es wird nicht geprüft, ob der Benutzer Zugriff auf den Namen der Zielwarteschlange hat.

Beispiel: Eine Aliaswarteschlange mit dem Namen PAYROLL.REQUEST wird in eine Zielwarteschlange von PAY.REQUEST aufgelöst. Wenn die Warteschlangensicherheit aktiv ist, müssen Sie nur über die Berechtigung zum Zugriff auf die Warteschlange PAYROLL.REQUEST berechtigt sein. Es wird keine Prüfung durchgeführt, um zu prüfen, ob Sie berechtigt sind, auf die Warteschlange PAY.REQUEST zuzugreifen.

z/OS *Verwenden von Aliaswarteschlangen zur Unterscheidung zwischen MQGET-und MQPUT-Anforderungen*

Der Bereich der MQI-Aufrufe, die in einer Zugriffsebene verfügbar sind, kann ein Problem verursachen, wenn Sie den Zugriff auf eine Warteschlange einschränken möchten, sodass nur der Aufruf **MQPUT** oder nur der Aufruf **MQGET** zulässig ist. Eine Warteschlange kann geschützt werden, indem zwei Aliasnamen definiert werden, die in diese Warteschlange aufgelöst werden: Eine Warteschlange, die es Anwendungen ermöglicht, Nachrichten aus der Warteschlange abzurufen, und eine, die es Anwendungen ermöglicht, Nachrichten in die Warteschlange zu stellen.

Der folgende Text enthält ein Beispiel dafür, wie Sie Ihre Warteschlangen für IBM MQ definieren können:

```
DEFINE QLOCAL(MUST_USE_ALIAS_TO_ACCESS) GET(ENABLED)
       PUT(ENABLED)

DEFINE QALIAS(USE_THIS_ONE_FOR_GETS) GET(ENABLED)
       PUT(DISABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)

DEFINE QALIAS(USE_THIS_ONE_FOR_PUTS) GET(DISABLED)
       PUT(ENABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)
```

Sie müssen außerdem die folgenden RACF-Definitionen vornehmen:

```
RDEFINE MQQUEUE hlq.MUST_USE_ALIAS_TO_ACCESS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_GETS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_PUTS UACC(NONE)
```

Dann stellen Sie sicher, dass keine Benutzer Zugriff auf die Warteschlange hlq.MUST_USE_ALIAS_TO_ACCESS haben, und geben Sie den entsprechenden Benutzern oder Gruppen Zugriff auf den Aliasnamen. Sie können dies mit den folgenden RACF-Befehlen tun:

```
PERMIT hlq.USE_THIS_ONE_FOR_GETS CLASS(MQQUEUE)
       ID(GETUSER,GETGRP) ACCESS(UPDATE)
PERMIT hlq.USE_THIS_ONE_FOR_PUTS CLASS(MQQUEUE)
       ID(PUTUSER,PUTGRP) ACCESS(UPDATE)
```

Dies bedeutet, dass die Benutzer-ID GETUSER und die Benutzer-IDs in der Gruppe GETGRP nur Nachrichten in der Aliaswarteschlange USE_THIS_ONE_FOR_GETS; über die Aliaswarteschlange MUST_USE_ALIAS_TO_ACCESS erhalten und die Benutzer-ID PUTUSER und die Benutzer-IDs in der Gruppe PUTGRP nur Nachrichten über die Aliaswarteschlange USE_THIS_ONE_FOR_PUTS einlegen dürfen.

Anmerkung:

1. Wenn Sie ein Verfahren wie dieses verwenden möchten, müssen Sie Ihre Anwendungsentwickler darüber informieren, dass sie ihre Programme entsprechend gestalten können.
2. Sie können ein solches Verfahren für die Zielwarteschlange verwenden, die Sie in einer MQSUB API-Anforderung bereitstellen, wenn Sie zwischen den Benutzern, die die Subskriptionen erstellen, und den Benutzern, die die Veröffentlichungen aus der Zielwarteschlange abrufen, unterscheiden möchten.

z/OS *Hinweise zu Modellwarteschlangen*

Um eine Modellwarteschlange zu öffnen, müssen Sie in der Lage sein, sowohl die Modellwarteschlange selbst als auch die dynamische Warteschlange, in die sie aufgelöst wird, zu öffnen. Definieren Sie generische RACF-Profile für dynamische Warteschlangen, einschließlich der vom IBM MQ-Dienstprogramm verwendeten dynamischen Warteschlangen.

Beim Öffnen einer Modellwarteschlange führt die IBM MQ-Sicherheit zwei Prüfungen zur Sicherheit einer Warteschlange durch:

1. Sind Sie berechtigt, auf die Modellwarteschlange zuzugreifen?
2. Sind Sie berechtigt, auf die dynamische Warteschlange zuzugreifen, in die die Modellwarteschlange aufgelöst wird?

Wenn der Name einer dynamischen Warteschlange einen abschließenden Stern (*) enthält, wird dieser Stern durch eine Zeichenfolge ersetzt, die von IBM MQ generiert wird, um eine dynamische Warteschlange mit einem eindeutigen Namen zu erstellen. Da jedoch der gesamte Name, einschließlich dieser generierten Zeichenfolge, zur Überprüfung der Berechtigung verwendet wird, sollten Sie generische Profile für diese Warteschlangen definieren.

Ein MQOPEN -Aufruf verwendet beispielsweise den Modellwarteschlangennamen CREDIT.CHECK.REPLY.MODEL und der dynamische Warteschlangenname CREDIT.REPLY.* auf Warteschlangenmanager (oder Gruppe mit gemeinsamer Warteschlange) MQSP

Dazu müssen Sie die folgenden RACF-Befehle zum Definieren der erforderlichen Warteschlangenprofile ausgeben:

```
RDEFINE MQQUEUE MQSP.CREDIT.CHECK.REPLY.MODEL
RDEFINE MQQUEUE MQSP.CREDIT.REPLY.**
```

Sie müssen außerdem die zugehörigen RACF PERMIT-Befehle ausgeben, um den Benutzerzugriff auf diese Profile zu ermöglichen.

Ein typischer dynamischer Warteschlangenname, der von einem MQOPEN erstellt wird, ist ähnlich wie CREDIT.REPLY.A346EF00367849A0. Der genaue Wert des letzten Qualifikationsmerkmals ist unvorhersehbar. Aus diesem Grund sollten Sie generische Profile für solche Warteschlangennamen verwenden.

Eine Reihe von IBM MQ-Dienstprogramme stellen Nachrichten in dynamische Warteschlangen. Sie sollten Profile für die folgenden Namen von dynamischen Warteschlangen definieren und RACF UPDATE den

Zugriff auf die relevanten Benutzer-IDs ermöglichen (siehe „Benutzer-IDs für die Sicherheitsprüfung unter z/OS" auf Seite 255 für die richtigen Benutzer-IDs):

```
SYSTEM.CSQUTIL.*  (used by CSQUTIL)
SYSTEM.CSQOREXX.* (used by the operations and control panels)
SYSTEM.CSQXCMD.*  (used by the channel initiator when processing CSQINPX)
CSQ4SAMP.*        (used by the IBM MQ supplied samples)
```

Sie können auch die Definition eines Profils in Betracht ziehen, um die Verwendung des Namens der dynamischen Warteschlange zu steuern, der standardmäßig in den Membern der Anwendungsprogrammierkopie verwendet wird. Die von IBM MQ bereitgestellten Copybooks enthalten den *DynamicQName*-Standardwert CSQ.*. Dadurch kann ein entsprechendes RACF-Profil eingerichtet werden.

Anmerkung: Verwenden Sie Anwendungsprogrammierer nicht, um einen Stern (*) für den Namen der dynamischen Warteschlange anzugeben. Wenn Sie dies tun, müssen Sie ein hlq. * * Profil in der MQQUEUE-Klasse und Sie müssen ihr einen umfassenden Zugriff erteilen. Dies bedeutet, dass dieses Profil auch für andere nicht dynamische Warteschlangen verwendet werden kann, die kein spezifischeres RACF-Profil besitzen. Ihre Benutzer könnten daher Zugriff auf Warteschlangen erhalten, auf die Sie nicht zugreifen möchten.

z/OS *Optionen in permanenten dynamischen Warteschlangen schließen*

Wenn eine Anwendung eine permanente dynamische Warteschlange öffnet, die von einer anderen Anwendung erstellt wurde, und versucht anschließend, diese Warteschlange mit der Option MQCLOSE zu löschen, werden beim Versuch einige zusätzliche Sicherheitsprüfungen angewendet.

Tabelle 33. Zugriffsebenen für Optionen zum Schließen von permanenten dynamischen Warteschlangen

MQCLOSE, Option	**RACF-Zugriffsebene, die für 'hlq.queuename' erforderlich ist**
MQCO_DELETE	ALTER
MQCO_DELETE_PURGE	ALTER

z/OS *Sicherheit und ferne Warteschlangen*

Wenn eine Nachricht in eine ferne Warteschlange gestellt wird, hängt die Warteschlangensicherheit, die vom lokalen Warteschlangenmanager implementiert wird, davon ab, wie die ferne Warteschlange beim Öffnen angegeben wird.

Es gelten die folgenden Regeln:

1. Wenn die ferne Warteschlange auf dem lokalen Warteschlangenmanager über den IBM MQ -Befehl DEFINE QREMOTE definiert wurde, ist die Warteschlange, die geprüft wird, der Name der fernen Warteschlange. Wenn z. B. eine ferne Warteschlange auf WS-Manager MQS1 wie folgt definiert ist:

   ```
   DEFINE QREMOTE(BANK7.CREDIT.REFERENCE)
          RNAME(CREDIT.SCORING.REQUEST)
          RQMNAME(BNK7)
          XMITQ(BANK1.TO.BANK7)
   ```

   In diesem Fall muss ein Profil für BANK7.CREDIT.REFERENCE in der Klasse MQQUEUE definiert sein.

2. Wenn der *ObjectQMgrName* für die Anforderung nicht in den lokalen Warteschlangenmanager aufgelöst wird, wird eine Sicherheitsprüfung für den aufgelösten (fernen) WS-Manager-Namen durchgeführt, außer im Fall einer Clusterwarteschlange, in der die Prüfung für den Namen der Clusterwarteschlange durchgeführt wird.

   Beispiel: Die Übertragungswarteschlange BANK1.TO.BANK7 ist in WS-Manager MQS1 definiert. Anschließend wird für MQS1 eine MQPUT1 -Anforderung ausgegeben, die *ObjectName* als BANK1.INTER-

BANK.TRANSFERS und eine *ObjectQMgrName* von BANK1.TO.BANK7 angibt. In diesem Fall muss der Benutzer, der die Anforderung ausführt, Zugriff auf die BANK1.TO.BANK7 haben.

3. Wenn Sie eine MQPUT -Anforderung in eine Warteschlange stellen und *ObjectQMgrName* als Aliasnamen des lokalen Warteschlangenmanagers angeben, wird nur der Warteschlangenname auf die Sicherheit überprüft, nicht die des Warteschlangenmanagers.

Wenn die Nachricht an den fernen WS-Manager kommt, kann sie möglicherweise einer zusätzlichen Sicherheitsverarbeitung unterliegen. Weitere Informationen finden Sie unter „Sicherheit für fernes Messaging” auf Seite 106.

z/OS *Sicherheit der Warteschlange für nicht zustellbare*

Besondere Hinweise gelten für die Warteschlange für dead-letter, da viele Benutzer in der Lage sein müssen, Nachrichten in die Warteschlange zu stellen, aber der Zugriff zum Abrufen von Nachrichten muss eng begrenzt sein. Dieses Ziel erreichen Sie, indem Sie für die Warteschlange für nicht zustellbare Nachrichten und eine Aliaswarteschlange unterschiedliche RACF-Berechtigungen anwenden.

Nicht zugestellte Nachrichten können in eine spezielle Warteschlange gestellt werden, die als "dead-letter queue" bezeichnet wird. Wenn Sie sensible Daten haben, die möglicherweise in dieser Warteschlange landen, müssen Sie die Sicherheitsauswirkungen in Betracht ziehen, da Sie nicht möchten, dass nicht berechtigte Benutzer diese Daten abrufen können.

Jede der folgenden Nachrichten muss in die Warteschlange für dead-letter gestellt werden dürfen:

- Anwendungsprogramme.
- Der Adressraum des Kanalinitiators und alle MCA-Benutzer-IDs. (Wenn das Profil RESLEVEL nicht vorhanden ist oder so definiert ist, dass Kanalbenutzer-IDs überprüft werden, benötigt die Kanalbenutzer-ID auch die Berechtigung zum Angeben von Nachrichten in die Warteschlange für dead-Mail.)
- CKTI, der von CICS bereitgestellte CICS-Taskinitiator.
- CSQQTRMN, der von IBM MQ bereitgestellte IMS-Auslösemonitor.

Die einzige Anwendung, die Nachrichten aus der Warteschlange für dead-letter (dead-letter) abrufen kann, sollte eine 'spezielle' Anwendung sein, die diese Nachrichten verarbeitet. Es tritt jedoch ein Problem auf, wenn Sie Anwendungen die RACF-Berechtigung UPDATE für die Warteschlange für nicht zustellbare Nachrichten für MQPUT-Anforderungen erteilen, da diese Nachrichten dann mithilfe von MQGET-Aufrufen automatisch aus der Warteschlange abrufen können. Sie können die Warteschlange für nicht zustellbare Nachrichten nicht für den Abruf von Operationen inaktivieren, da Sie, wenn Sie dies tun, nicht einmal die 'speziellen' Anwendungen abrufen können.

Eine Lösung für dieses Problem ist die Einrichtung eines zweistufigen Zugriffs auf die Warteschlange für dead-letter. CKTI-, Nachrichten-Channel-Agent-Transaktionen oder der Adressraum des Kanalinitiators und 'spezielle' Anwendungen haben direkten Zugriff; andere Anwendungen können nur über eine Aliaswarteschlange auf die Warteschlange für dead-letter zugreifen. Dieser Aliasname ist definiert, damit Anwendungen Nachrichten in die Warteschlange für nicht zustellbare Nachrichten einlegen können, aber keine Nachrichten von dieser Warteschlange erhalten.

So könnte es funktionieren:

1. Definieren Sie die echte dead-letter-Warteschlange mit den Attributen PUT (ENABLED) und GET (ENABLED), wie in der Beispieldatei thlqual.SCSQPROC (CSQ4INYG) dargestellt.
2. Erteilen Sie den folgenden Benutzer-IDs die RACF-Berechtigung UPDATE für die Warteschlange für nicht zustellbare Nachrichten:
   - Benutzer-IDs, unter denen der CKTI-und der MCAs-oder Kanalinitiatoradressraum ausgeführt werden.
   - Die Benutzer-IDs, die der 'speziellen' -Warteschlange für die Verarbeitung von Nachrichten in der Warteschlange zugeordnet sind.
3. Definieren Sie eine Aliaswarteschlange, die in die echte Warteschlange für dead-letter aufgelöst wird, geben Sie jedoch die folgenden Attribute für die Aliaswarteschlange an: PUT (ENABLED) und GET (DISABLED). Geben Sie der Aliaswarteschlange einen Namen mit demselben Stamm wie der Name der

Warteschlange für den Namen der Warteschlange an, aber hängen Sie die Zeichen ". PUT" an diesen Stamm an. Wenn der Name der Warteschlange für dead-letter beispielsweise hlq.DEAD.QUEUE lautet, würde der Name der Aliaswarteschlange hlq.DEAD.QUEUE.PUT.

4. Um eine Nachricht in die Warteschlange für dead-letter zu stellen, verwendet eine Anwendung die Aliaswarteschlange. Dies ist die, die Ihre Anwendung ausführen muss:
   - Rufen Sie den Namen der echten Warteschlange für dead-letter ab. Dazu wird das WS-Manager-Objekt mit `MQOPEN` geöffnet und gibt dann einen `MQINQ` aus, um den Namen der Warteschlange für dead-letter abzurufen.
   - Erstellen Sie den Namen der Aliaswarteschlange, indem Sie die Zeichen '.PUT' an diesen Namen anfügen, in diesem Fall hlq.DEAD.QUEUE.PUT.
   - Öffnen Sie die Aliaswarteschlange hlq.DEAD.QUEUE.PUT.
   - Stellen Sie die Nachricht in die echte Warteschlange für dead-letter, indem Sie eine `MQPUT` -Operation für die Aliaswarteschlange ausgeben.
5. Erteilen Sie der Benutzer-ID, die der Anwendung zugeordnet ist, die RACF-Berechtigung UPDATE für die Aliaswarteschlange, aber keine Zugriffsberechtigung (Berechtigung NONE) für die tatsächliche Warteschlange für nicht zustellbare Nachrichten. Dies bedeutet Folgendes:
   - Die Anwendung kann Nachrichten mithilfe der Aliaswarteschlange in die Warteschlange für dead-letter stellen.
   - Die Anwendung kann Nachrichten aus der Warteschlange für nicht zustellbare Nachrichten nicht mit Hilfe der Aliaswarteschlange abrufen, da die Aliaswarteschlange für get-Operationen inaktiviert ist.

   Die Anwendung kann keine Nachrichten aus der tatsächlichen Warteschlange für nicht zustellbare Nachrichten abrufen, da sie nicht über die korrekte RACF-Berechtigung verfügt.

Tabelle 34 auf Seite 221 fasst die RACF-Berechtigungen zusammen, die für die verschiedenen Teilnehmer in dieser Lösung erforderlich sind.

Tabelle 34. RACF-Berechtigung für die Warteschlange für nicht zustellbare Nachrichten und den zugehörigen Aliasnamen

Zugeordnete Benutzer-IDs	**Reale Warteschlange für dead-letter (hlq.DEAD.QUEUE)**	**Warteschlange für Aliasnamendeadletter (HLQ.DEAD.QUEUE.PUT)**
MCA-oder Kanalinitiatoradressraum und CKTI	UPDATE	KEINE
'Sonderanwendung' (für die Verarbeitung von Nachrichten in der Warteschlange für dead-letter)	UPDATE	KEINE
Benutzerspezifische Benutzer-IDs für Anwendungen	KEINE	UPDATE

Wenn Sie diese Methode verwenden, kann die Anwendung die maximale Nachrichtenlänge (MAXMSGL) der Warteschlange für dead-Mail nicht ermitteln. Dies liegt daran, dass das Attribut MAXMSGL nicht aus einer Aliaswarteschlange abgerufen werden kann. Daher wird in Ihrer Anwendung vorausgesetzt, dass die maximale Nachrichtenlänge 100 MB beträgt, die maximale von IBM MQ for z/OS unterstützte Größe. Die Warteschlange für echte dead-letter sollte auch mit einem MAXMSGL-Attribut von 100 MB definiert werden.

Anmerkung: Benutzerdefinierte Anwendungsprogramme verwenden normalerweise nicht die alternative Benutzerberechtigung, um Nachrichten in die Warteschlange für dead-letter zu stellen. Dadurch wird die Anzahl der Benutzer-IDs, die Zugriff auf die Warteschlange für dead-letter haben, reduziert.

z/OS *Sicherheit der Systemwarteschlange*

Sie müssen den RACF-Zugriff einrichten, damit bestimmte Benutzer-IDs auf bestimmte Systemwarteschlangen zugreifen können.

Auf viele Systemwarteschlangen wird durch die Nebenkomponenten von IBM MQ zugegriffen:

- Das Dienstprogramm CSQUTIL
- Das Dienstprogramm für die Nachrichtensicherheitsrichtlinie (CSQ0UTIL)
- Die Operationen und Steuerkonsolen
- Der Adressraum des Kanalinitiators (einschließlich des Pub/Sub-Dämons in der Warteschlange)
- Der mqweb-Server, der von der MQ Console und dem REST API verwendet wird.

Die Benutzer-IDs, unter denen diese Komponenten ausgeführt werden, müssen RACF-Zugriff auf diese Warteschlangen erhalten, wie in Tabelle 35 auf Seite 222 gezeigt.

Tabelle 35. Erforderlicher Zugriff auf die SYSTEM-Warteschlangen durch IBM MQ

SYSTEM-Warteschlange	CSQUTIL	CSQ0UTIL	mqweb-Server	Operations-und Steuerkonsolen	Kanalinitiator für verteilte Steuerung von Warteschlangen
SYSTEM.ADMIN.CHANNEL.EVENT	-	-	-	-	UPDATE
SYSTEM.ADMIN.COMMAND.QUEUE	-	-	UPDATE	-	-
SYSTEM.BROKER.ADMIN.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.CONTROL.QUEUE	-	-	-	-	ALTER
SYSTEM.BROKER.DEFAULT.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	-	-	-	-	UPDATE
SYSTEM.CHANNEL.INITQ	-	-	-	-	UPDATE
SYSTEM.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.CLUSTER.COMMAND.QUEUE	-	-	-	-	ALTER
SYSTEM.CLUSTER.REPOSITORY.QUEUE	-	-	-	-	UPDATE
SYSTEM.CLUSTER.TRANSMIT.QUEUE	-	-	-	-	ALTER
SYSTEM.COMMAND.INPUT	UPDATE	-	-	UPDATE	UPDATE
SYSTEM.COMMAND.REPLY.*	-	-	-	-	UPDATE
SYSTEM.COMMAND.REPLY.MODEL	UPDATE	-	-	UPDATE	UPDATE
SYSTEM.CSQOREXX.*	-	-	-	UPDATE	-
SYSTEM.CSQUTIL.*	UPDATE	-	-	-	-
SYSTEM.CSQXCMD.*	-	-	-	-	UPDATE
SYSTEM.HIERARCHY.STATE	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.CONTROL	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.PUBS	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.FANREQ	-	-	-	-	UPDATE
SYSTEM.PROTECTION.ERROR.QUEUE	-	-	-	-	UPDATE

Tabelle 35. Erforderlicher Zugriff auf die SYSTEM-Warteschlangen durch IBM MQ (Forts.)					
SYSTEM-Warteschlange	**CSQUTIL**	**CSQ0UTIL**	**mqweb-Server**	**Operations-und Steuerkonsolen**	**Kanalinitiator für verteilte Steuerung von Warteschlangen**
SYSTEM.PROTECTION.POLICY.QUEUE	-	UPDATE „1” auf Seite 223	-	-	READ
SYSTEM.QSG.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.QSG.TRANSMIT.QUEUE	-	-	-	-	UPDATE
SYSTEM.REST.REPLY.QUEUE	-	-	UPDATE	-	-
SYSTEM.BLUEMIX.REGISTRATION.QUEUE	-	-	-	-	UPDATE

Anmerkungen:

1. Der Benutzer des Advanced Message Security-Adressraums benötigt ebenfalls Lesezugriff für diese Warteschlange.

z/OS *API-Ressourcenzugriffsschutz-Kurzreferenz*

Eine Zusammenfassung der Optionen **MQOPEN**, **MQPUT1**, **MQSUB**und **MQCLOSE** sowie des Zugriffs, der für die verschiedenen Ressourcensicherheitstypen erforderlich ist.

Tabelle 36. MQOPEN-, MQPUT1-, MQSUB-und MQCLOSE-Optionen und die erforderliche Sicherheitsberechtigung. Callouts, die wie diese **(1)** angezeigt werden, beziehen sich auf die Anmerkungen, die auf diese Tabelle folgen.				
	Erforderliche Mindestzugriffsebene für RACF			
RACF-Klasse:	**MXTOPIC**	**MQQUEUE oder MXQUEUE(1)**	**MQADMIN oder MXADMIN**	**MQADMIN oder MXADMIN**
RACF-Profil:	**(15 oder 16)**	**(2)**	**(3)**	**(4)**
Option MQOPEN				
MQOO_INQUIRE		READ **(5)**	Keine Prüfung	Keine Prüfung
MQOO_BROWSE		READ	Keine Prüfung	Keine Prüfung
MQOO_INPUT_*		UPDATE	Keine Prüfung	Keine Prüfung
MQOO_SAVE_ALL_CONTEXT **(6)**		UPDATE	Keine Prüfung	Keine Prüfung
MQOO_OUTPUT (USAGE = NORMAL) **(7)**		UPDATE	Keine Prüfung	Keine Prüfung
MQOO_PASS_IDENTITY_CONTEXT **(8)**		UPDATE	READ	Keine Prüfung
MQOO_PASS_ALL_CONTEXT **(8) (9)**		UPDATE	READ	Keine Prüfung
MQOO_SET_IDENTITY_CONTEXT **(8) (9)**		UPDATE	UPDATE	Keine Prüfung
MQOO_SET_ALL_CONTEXT **(8) (10)**		UPDATE	STEUERUNG	Keine Prüfung
MQOO_OUTPUT (USAGE (XMITQ) **(11)**		UPDATE	STEUERUNG	Keine Prüfung
MQOO_OUTPUT (Themenobjekt)	UPDATE **(16)**			

Tabelle 36. MQOPEN-, MQPUT1-, MQSUB-und MQCLOSE-Optionen und die erforderliche Sicherheitsberechtigung. Callouts, die wie diese **(1)** angezeigt werden, beziehen sich auf die Anmerkungen, die auf diese Tabelle folgen. *(Forts.)*

	Erforderliche Mindestzugriffsebene für RACF			
RACF-Klasse:	**MXTOPIC**	**MQQUEUE oder MXQUEUE(1)**	**MQADMIN oder MXADMIN**	**MQADMIN oder MXADMIN**
RACF-Profil:	**(15 oder 16)**	**(2)**	**(3)**	**(4)**
MQOO_OUTPUT (Aliaswarteschlange für Themenobjekt)	UPDATE **(16)**	UPDATE		
MQOO_SET		ALTER	Keine Prüfung	Keine Prüfung
MQOO_ALTERNATE_USER_AUTHORITY		**(12)**	**(12)**	UPDATE
Option MQPUT1				
In normale Warteschlange einreihen **(7)**		UPDATE	Keine Prüfung	Keine Prüfung
MQPMO_PASS_IDENTITY_CONTEXT		UPDATE	READ	Keine Prüfung
MQPMO_PASS_ALL_CONTEXT		UPDATE	READ	Keine Prüfung
MQPMO_SET_IDENTITY_CONTEXT		UPDATE	UPDATE	Keine Prüfung
MQPMO_SET_ALL_CONTEXT		UPDATE	STEUERUNG	Keine Prüfung
MQOO_OUTPUT In eine Übertragungswarteschlange einreihen **(11)**		UPDATE	STEUERUNG	Keine Prüfung
MQOO_OUTPUT (Themenobjekt)	UPDATE **(16)**			
MQOO_OUTPUT (Aliaswarteschlange für Themenobjekt)	UPDATE **(16)**	UPDATE		
MQPMO_ALTERNATE_USER_AUTHORITY		**(13)**	**(13)**	UPDATE
Option MQCLOSE				
MQCO_DELETE **(14)**		ALTER	Keine Prüfung	Keine Prüfung
MQCO_DELETE_PURGE **(14)**		ALTER	Keine Prüfung	Keine Prüfung
MQCO_REMOVE_SUB	ALTER **(15)**			
Option MQSUB				
MQSO_CREATE	ALTER **(15)**	**(17)**	**(18)**	
MQSO_ALTER	ALTER **(15)**	**(17)**	**(18)**	
MQSO_RESUME	READ **(15)**	**(17)**	Keine Prüfung	
MQSO_ALTERNATE_USER_AUTHORITY				UPDATE
MQSO_SET_IDENTITY_CONTEXT			**(18)**	

Anmerkung:

1. Diese Option ist nicht auf Warteschlangen beschränkt. Verwenden Sie die Klasse MQNLIST oder MXNLIST für Namenslisten und die MQPROC-oder MXPROC-Klasse für Prozesse.
2. Verwenden Sie das RACF-Profil: hlq.resourcename

3. Verwenden Sie das RACF-Profil: hlq.CONTEXT.queuename
4. Verwenden Sie das Profil RACF: hlq.ALTERNATE.USER. `alternateuserid`

   `alternateuserid` ist die Benutzer-ID, die im Feld *`AlternateUserId`* des Objektdeskriptors angegeben ist. Beachten Sie, dass für diese Prüfung bis zu 12 Zeichen des Feldes *`AlternateUserId`* verwendet werden, im Gegensatz zu anderen Prüfungen, bei denen nur die ersten 8 Zeichen einer Benutzer-ID verwendet werden.
5. Beim Öffnen des Warteschlangenmanagers für Anfragen wird keine Prüfung durchgeführt.
6. MQOO_INPUT_ * muss auch angegeben werden. Dies ist für eine lokale, Modell-oder Aliaswarteschlange gültig.
7. Diese Prüfung wird für eine lokale oder Modellwarteschlange durchgeführt, die ein Warteschlangenattribut **Usage** von MQUS_NORMAL und auch einen Aliasnamen oder eine ferne Warteschlange enthält (der für den verbundenen Warteschlangenmanager definiert ist).) Wenn es sich bei der Warteschlange um eine ferne Warteschlange handelt, die explizit eine *`ObjectQMgrName`* (nicht den Namen des verbundenen Warteschlangenmanagers) angibt, wird die Prüfung für die Warteschlange mit demselben Namen wie *`ObjectQMgrName`* ausgeführt (wobei es sich um eine lokale Warteschlange mit einem Warteschlangenattribut von **Usage** MQUS_TRANSMISSION handeln muss).
8. MQOO_OUTPUT muss auch angegeben werden.
9. MQOO_PASS_IDENTITY_CONTEXT wird auch durch diese Option impliziert.
10. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT und MQOO_SET_IDENTITY_CONTEXT werden auch durch diese Option impliziert.
11. Diese Prüfung wird für eine lokale oder Modellwarteschlange durchgeführt, die über ein Warteschlangenattribut **Usage** von MQUS_TRANSMISSION verfügt und direkt für die Ausgabe geöffnet wird. Sie gilt nicht, wenn eine ferne Warteschlange geöffnet wird.
12. Es muss mindestens ein MQOO_INQUIRE-, MQOO_BROWSE-, MQOO_INPUT_ *-, MQOO_OUTPUT- oder MQOO_SET-Wert angegeben werden. Die durchgeführte Prüfung ist die gleiche wie die für die anderen angegebenen Optionen.
13. Die durchgeführte Prüfung ist die gleiche wie die für die anderen angegebenen Optionen.
14. Dies gilt nur für permanente dynamische Warteschlangen, die direkt geöffnet wurden, d. a. nicht über eine Modellwarteschlange geöffnet wurden. Es ist keine Sicherheit erforderlich, um eine temporäre dynamische Warteschlange zu löschen.
15. Verwenden Sie das RACF-Profil hlq.SUBSCRIBE.topicname.
16. Verwenden Sie das RACF-Profil hlq.PUBLISH.topicname.
17. Wenn Sie in der MQSUB-Anforderung eine Zielwarteschlange für die zu sensenden Veröffentlichungen angegeben haben, wird eine Sicherheitsprüfung für diese Warteschlange ausgeführt, um sicherzustellen, dass Sie die Berechtigung für diese Warteschlange angegeben haben.
18. Wenn in der MQSUB-Anforderung mit MQSO_CREATE-oder MQSO_ALTER-Optionen ein beliebtes Identitätskontextfeld in der MQSD-Struktur festgelegt werden soll, müssen Sie auch die Option MQSO_SET_IDENTITY_CONTEXT angeben, und Sie benötigen außerdem die entsprechende Berechtigung für das Kontextprofil für die Zielwarteschlange.

z/OS *Profile für Themensicherheit*

Wenn die Themensicherheit aktiv ist, müssen Sie Profile in den entsprechenden Klassen definieren und die erforderlichen Gruppen oder Benutzer-IDs auf diese Profile zugreifen lassen.

Das Konzept der Themensicherheit in einer Themenstruktur wird im Abschnitt Publish/Subscribe-Sicherheit beschrieben.

Wenn die Themensicherheit aktiv ist, müssen Sie die folgenden Aktionen ausführen:

- Definieren Sie Profile in den Klassen **MXTOPIC** oder **GMXTOPIC** .
- Erlauben Sie den erforderlichen Gruppen oder Benutzer-IDs Zugriff auf diese Profile, damit sie IBM MQ-API-Anforderungen ausgeben können, die Themen verwenden.

Profile für die Themensicherheit haben das folgende Format:

```
hlq.SUBSCRIBE.topicname
hlq.PUBLISH.topicname
```

Dabei gilt Folgendes:

- `hlq` ist entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange).
- `topicname` ist der Name des Topic-Verwaltungsknotens in der Themenstruktur, der entweder dem Thema zugeordnet ist, das über einen MQSUB-Aufruf subskribiert wird, oder über einen MQOPEN-Aufruf veröffentlicht wird.

Ein Profil, dem der Name des Warteschlangenmanagers vorangesetzt ist, steuert den Zugriff auf ein einzelnes Thema in diesem Warteschlangenmanager. Ein Profil, das den Namen der Gruppe mit gemeinsamer Warteschlange als Präfix hat, steuert den Zugriff auf ein oder mehrere Themen mit diesem Themennamen auf allen Warteschlangenmanagern innerhalb der Gruppe mit gemeinsamer Warteschlange. Dieser Zugriff kann auf einem einzelnen WS-Manager überschrieben werden, indem ein WS-Managerebenenprofil für dieses Thema in diesem WS-Manager definiert wird.

Wenn Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit auf der Ebene des Warteschlangenmanagers als auch die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, sucht IBM MQ zuerst nach einem Profil, das dem Namen des Warteschlangenmanagers vorangestellt ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist.

Abonnieren

Um ein Thema zu subskribieren, benötigen Sie Zugriff sowohl auf das Thema, das Sie subskribieren möchten, als auch auf die Zielwarteschlange für die Veröffentlichungen.

Wenn Sie eine MQSUB-Anforderung absetzen, werden die folgenden Sicherheitsprüfungen durchgeführt:

- Ob Sie über die entsprechende Zugriffsebene zum Subskribieren dieses Themas verfügen und ob die Zielwarteschlange (falls angegeben) für die Ausgabe geöffnet wird
- Gibt an, ob Sie über die entsprechende Zugriffsebene für diese Zielwarteschlange verfügen.

Tabelle 37. Erforderliche Zugriffsebene für Topic-Sicherheit zum Abonnieren

MQSUB, Option	**RACF -Zugriff erforderlich für Profil `hlq.SUBSCRIBE.topicname` in MXTOPIC-Klasse**
MQSO_CREATE und MQSO_ALTER	ALTER
MQSO_RESUME	READ

Tabelle 38. Zusätzliche Berechtigung, die erforderlich ist, um eine nicht verwaltete Zielwarteschlange zu subskribieren

MQSUB, Option	**RACF -Zugriff erforderlich für das Profil `hlq.CONTEXT.queuename` in der Klasse MQADMIN oder MXADMIN**
MQSO_CREATE, MQSO_ALTER und MQSO_RESUME	UPDATE
	RACF -Zugriff erforderlich für Profil `hlq.queuename` in MQQUEUE-oder MXQUEUE-Klasse
MQSO_CREATE und MQSO_ALTER	UPDATE

Tabelle 38. Zusätzliche Berechtigung, die erforderlich ist, um eine nicht verwaltete Zielwarteschlange zu subskribieren (Forts.)	
MQSUB, Option	**RACF -Zugriff erforderlich für das Profil `hlq.CONTEXT.queuename` in der Klasse MQADMIN oder MXADMIN**
	RACF -Zugriff erforderlich für Profil `hlq.ALTERNATE.USER.alternateuserid` in Klasse MQADMIN oder MXADMIN
MQSO_ALTERNATE_USER_AUTHORITY	UPDATE

Hinweise zu verwalteten Warteschlangen für Subskriptionen

Es wird eine Sicherheitsprüfung durchgeführt, um zu prüfen, ob Sie das Thema subskribieren können. Es werden jedoch keine Sicherheitsprüfungen ausgeführt, wenn die verwaltete Warteschlange erstellt wird, oder Sie können feststellen, ob Sie Zugriff auf diese Zielwarteschlange haben, um Nachrichten in diese Warteschlange zu stellen.

Es ist nicht möglich, eine verwaltete Warteschlange zu schließen.

Die verwendeten Modellwarteschlangen sind: `SYSTEM.DURABLE.MODEL.QUEUE` und `SYSTEM.NDURABLE.MODEL.QUEUE`.

Die aus diesen Modellwarteschlangen erstellten verwalteten Warteschlangen haben das Format `SYSTEM.MANAGED.DURABLE.A346EF00367849A0` und `SYSTEM.MANAGED.NDURABLE.A346EF0036785EA0` , wobei das letzte Qualifikationsmerkmal unvorhersehbar ist.

Geben Sie keinen Benutzerzugriff auf diese Warteschlangen an. Die Warteschlangen können mit generischen Profilen des Formulars `SYSTEM.MANAGED.DURABLE.*` und `SYSTEM.MANAGED.NDURABLE.*` geschützt werden, ohne dass Berechtigungen erteilt werden.

Nachrichten können über die Kennung, die in der MQSUB-Anforderung zurückgegeben wird, aus diesen Warteschlangen abgerufen werden.

Wenn Sie explizit einen MQCLOSE-Aufruf für eine Subskription mit der angegebenen Option MQCO_REMOVE_SUB absetzen und die Subskription, die Sie unter dieser Kennung schließen, nicht erstellt haben, wird zum Zeitpunkt des Abschlusses eine Sicherheitsprüfung ausgeführt, um sicherzustellen, dass Sie über die korrekte Berechtigung zum Ausführen der Operation verfügen.

Tabelle 39. Zugriffsebene, die für Profile für die Topic-Sicherheit zum Schließen einer Subskriptionsoperation erforderlich ist	
MQCLOSE, Option	**RACF -Zugriff erforderlich für Profil `hlq.SUBSCRIBE.topicname` in MXTOPIC-Klasse**
MQCO_REMOVE_SUB	ALTER

Veröffentlichen

Zum Publizieren in einem Thema benötigen Sie Zugriff auf das Thema und, wenn Sie Aliaswarteschlangen verwenden, auch in die Aliaswarteschlange.

Tabelle 40. Zugriffsebene, die für die zu veröffentlichende Themensicherheit erforderlich ist	
MQOPEN-oder MQPUT1-Option	**RACF -Zugriff erforderlich für Profil `hlq.PUBLISH.topicname` in MXTOPIC-Klasse**
MQOO_OUTPUT oder MQPUT1	UPDATE

Tabelle 41. Zugriffsebene, die zum Öffnen einer Aliaswarteschlange erforderlich ist, die in ein Thema aufgelöst wird	
MQOPEN-oder MQPUT1-Option	**RACF -Zugriff erforderlich für das Profil `hlq.queuename` in der Klasse MQQUEUE oder MXQUEUE für die Aliaswarteschlange**
MQOO_OUTPUT oder MQPUT1	UPDATE

Weitere Informationen für die Durchführung der Themensicherheit, wenn eine Aliaswarteschlange zur Auflösung eines Themennamens für die Veröffentlichung geöffnet ist, finden Sie im Abschnitt „Überlegungen zu Aliaswarteschlangen, die in Themen für eine Veröffentlichungsoperation aufgelöst werden” auf Seite 228.

Informationen zur Verwendung von Aliaswarteschlangen für Zielwarteschlangen für PUT- oder GET-Einschränkungen finden Sie unter „Hinweise zu Aliaswarteschlangen” auf Seite 217.

Wenn die RACF-Zugriffsebene, die eine Anwendung für ein Themensicherheitsprofil hat, geändert wird, werden die Änderungen nur für neue Objektkennungen (also ein neuer MQSUB- oder MQOPEN-Aufruf) angewendet, die für dieses Thema abgerufen werden. Diese Kennungen, die zum Zeitpunkt der Änderung bereits vorhanden sind, behalten ihren bereits vorhandenen Zugriff auf das Thema bei. Außerdem behalten die vorhandenen Subskribenten ihren Zugriff auf alle Subskriptionen vor, die sie bereits erstellt haben.

Überlegungen zu Aliaswarteschlangen, die in Themen für eine Veröffentlichungsoperation aufgelöst werden

Wenn Sie einen MQOPEN- oder MQPUT1-Aufruf für eine Aliaswarteschlange ausgeben, die in ein Thema aufgelöst wird, führt IBM MQ zwei Ressourcenprüfungen aus:

- Der erste Parameter für den Aliaswarteschlangennamen, der im Objektdeskriptor (MQOD) im MQOPEN- oder MQPUT1-Aufruf angegeben wurde.
- Die zweite mit dem Thema, in das die Aliaswarteschlange aufgelöst wird.

Sie müssen sich bewusst sein, dass dieses Verhalten sich von dem Verhalten unterscheidet, das Sie erhalten, wenn Aliaswarteschlangen in andere Warteschlangen aufgelöst werden. Sie benötigen den richtigen Zugriff auf beide Profile, damit die Veröffentlichungsaktion fortgesetzt werden kann.

Sicherheit des Systemthemas

Auf die folgenden Systemthemen wird über den Adressraum des Kanalinitiators zugegriffen.

Die Benutzer-IDs, unter denen diese ausgeführt wird, müssen RACF -Zugriff auf diese Warteschlangen erhalten (siehe Tabelle 42 auf Seite 228).

Tabelle 42. Erforderlicher Zugriff auf die Themen zu SYSTEM		
SYSTEM-Thema	**Profil**	**Kanalinitiator für verteilte Steuerung von Warteschlangen**
SYSTEM.BROKER.ADMIN.STREAM	hlq.PUBLISH.topicname	UPDATE
SYSTEM.BROKER.ADMIN.STREAM	hlq.SUBSCRIBE.topicname	ALTER

z/OS *Profile für Prozesse*

Wenn die Prozesssicherheit aktiv ist, müssen Sie Profile in den entsprechenden Klassen definieren und die erforderlichen Gruppen oder Benutzer-IDs Zugriff auf diese Profile zulassen.

Wenn die Prozesssicherheit aktiv ist, müssen Sie:

- Definieren Sie Profile in den Klassen **MQPROC** oder **GMQPROC** , wenn Sie Profile in Großbuchstaben verwenden.
- Definieren Sie Profile in den Klassen **MXPROC** oder **GMXPROC** , wenn Sie Profile mit Groß-/Kleinschreibung verwenden.
- Erteilen Sie den erforderlichen Gruppen oder Benutzer-IDs Zugriff auf diese Profile, damit sie IBM MQ-API-Anforderungen ausgeben können, die Prozesse verwenden.

Profile für Prozesse haben das folgende Format:

```
hlq.processname
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein und `processname` ist der Name des Prozesses, der geöffnet wird.

Ein Profil, das dem Namen des Warteschlangenmanagers vorangesetzt ist, steuert den Zugriff auf eine einzelne Prozessdefinition in diesem Warteschlangenmanager. Ein Profil, das den Namen der Gruppe mit gemeinsamer Warteschlange als Präfix hat, steuert den Zugriff auf eine oder mehrere Prozessdefinitionen für alle Warteschlangenmanager mit diesem Namen innerhalb der Gruppe mit gemeinsamer Warteschlange. Dieser Zugriff kann auf einem einzelnen WS-Manager außer Kraft gesetzt werden, indem ein WS-Managerebenenprofil für diese Prozessdefinition in diesem Warteschlangenmanager definiert wird.

Wenn Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit auf der Ebene des Warteschlangenmanagers als auch die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, sucht IBM MQ zuerst nach einem Profil, das dem Namen des Warteschlangenmanagers vorangestellt ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist.

In der folgenden Tabelle ist der Zugriff dargestellt, der für das Öffnen eines Prozesses erforderlich ist.

Tabelle 43. Zugriffsebenen für die Prozesssicherheit

MQOPEN, Option	**Erforderliche RACF-Zugriffsebene für hlq.processname**
MQOO_INQUIRE	READ

Beispiel: Auf dem Queue Manager MQS9 muss die RACF-Gruppe INQVPRC in der Lage sein, (`MQINQ`) bei allen Prozessen, die mit dem Buchstaben V beginnen, abzufragen. Die Definitionen für RACF lauten wie folgt:

```
RDEFINE MQPROC MQS9.V* UACC(NONE)
PERMIT MQS9.V* CLASS(MQPROC) ID(INQVPRC) ACCESS(READ)
```

Die alternative Benutzersicherheit kann auch aktiv sein, abhängig von den Optionen zum Öffnen, die beim Öffnen eines Prozessdefinitionsobjekts angegeben werden.

z/OS *Profile für Namenslisten*

Wenn die Namenslistensicherheit aktiv ist, definieren Sie Profile in den entsprechenden Klassen und geben den erforderlichen Gruppen oder Benutzer-IDs Zugriff auf diese Profile.

Wenn die Namenslistensicherheit aktiv ist, müssen Sie:

- Definieren Sie Profile in den Klassen **MQNLIST** oder **GMQNLIST** , wenn Sie Profile in Großbuchstaben verwenden.
- Definieren Sie Profile in den Klassen **MXNLIST** oder **GMXNLIST** , wenn Sie Profile mit Groß-/Kleinschreibung verwenden.
- Geben Sie die erforderlichen Gruppen oder Benutzer-IDs für diese Profile an.

Profile für namelists haben das folgende Format:

```
hlq.namelistname
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein und `namelistname` ist der Name der Namensliste, die geöffnet wird.

Ein Profil, dem der Name des Warteschlangenmanagers vorangesetzt ist, steuert den Zugriff auf eine einzige Namensliste auf diesem Warteschlangenmanager. Ein Profil, das den Namen der Gruppe mit gemeinsamer Warteschlange als Präfix hat, steuert den Zugriff auf eine oder mehrere Namenslisten mit diesem Namen auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange. Dieser Zugriff kann auf einem einzelnen WS-Manager außer Kraft gesetzt werden, indem ein Profil auf WS-Managerebene für diese Namensliste auf diesem Warteschlangenmanager definiert wird.

Wenn Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit auf der Ebene des Warteschlangenmanagers als auch die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, sucht IBM MQ zuerst nach einem Profil, das dem Namen des Warteschlangenmanagers vorangestellt ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist.

In der folgenden Tabelle ist der Zugriff dargestellt, der zum Öffnen einer Namensliste erforderlich ist.

Tabelle 44. Zugriffsebenen für die Sicherheit von Namenslisten

MQOPEN, Option	**RACF-Zugriffsebene, die für 'hlq.namelistname' erforderlich ist**
MQOO_INQUIRE	READ

Beispiel: Im Warteschlangenmanager (oder in der Gruppe mit gemeinsamer Warteschlange) PQM3 muss die RACF-Gruppe DEPT571 den Wert (`MQINQ`) in den folgenden Namenslisten abfragen können:

- Alle Namenslisten, die mit " DEPT571 " beginnen.
- PRINTER/DESTINATIONS/DEPT571
- AGENCY/REQUEST/QUEUES
- WAREHOUSE.BROADCAST

Die entsprechenden RACF-Definitionen lauten wie folgt:

```
RDEFINE MQNLIST PQM3.DEPT571.** UACC(NONE)
PERMIT PQM3.DEPT571.** CLASS(MQNLIST) ID(DEPT571) ACCESS(READ)

RDEFINE GMQNLIST NLISTS.FOR.DEPT571 UACC(NONE)
        ADDMEM(PQM3.PRINTER/DESTINATIONS/DEPT571,
               PQM3.AGENCY/REQUEST/QUEUES,
               PQM3.WAREHOUSE.BROADCAST)
PERMIT NLISTS.FOR.DEPT571 CLASS(GMQNLIST) ID(DEPT571) ACCESS(READ)
```

Die alternative Benutzersicherheit ist möglicherweise aktiv, abhängig von den Optionen, die beim Öffnen eines Namenslistenobjekts angegeben wurden.

Sicherheit der Systemnamensliste

Der Zugriff auf viele der Systemnamenslisten erfolgt über die Nebenkomponenten von IBM MQ:

- Das Dienstprogramm CSQUTIL
- Die Operationen und Steuerkonsolen
- Der Adressraum des Kanalinitiators (einschließlich des Dämons in der Warteschlange für eingereimtes Publish/Subscribe)

Die Benutzer-IDs, unter denen diese ausgeführt werden, müssen RACF -Zugriff auf diese Namenslisten erhalten, wie in Tabelle 45 auf Seite 231dargestellt.

Tabelle 45. Für IBM MQ erforderlicher Zugriff auf die SYSTEM-Namenslisten

SYSTEM-Namensliste	CSQUTIL	Operations- und Steuer-konsolen	Kanalinitiator für verteilte Steue-rung von War-teschlangen
SYSTEM.QPUBSUB.QUEUE.NAMELIST	-	-	READ
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	-	-	READ

z/OS *Profile für alternative Benutzersicherheit*

Wenn die alternative Benutzersicherheit aktiv ist, müssen Sie Profile in den entsprechenden Klassen definieren und die erforderlichen Gruppen oder Benutzer-IDs auf diese Profile zugreifen lassen.

Weitere Informationen zu *AlternateUserId*finden Sie unter AlternateUserID (MQCHAR12).

Wenn die alternative Benutzersicherheit aktiv ist, müssen Sie:

- Definieren Sie Profile in den Klassen 'MQADMIN' oder 'GMQADMIN', wenn Sie Profile in Großbuchstaben verwenden.
- Definieren Sie Profile in den Klassen "MXADMIN" oder "GMXADMIN", wenn Sie Profile mit Groß-/Kleinschreibung verwenden.

Die erforderlichen Gruppen oder Benutzer-IDs können auf diese Profile zugreifen, so dass sie die Optionen ALTERNATE_USER_AUTHORITY verwenden können, wenn das Objekt geöffnet wird.

Profile für eine Sicherheit mit alternativen Benutzer-IDs können auf Subsystemebene oder auf Ebene der Gruppe mit gemeinsamer Warteschlange angegeben werden und haben das folgende Format:

```
hlq.ALTERNATE.USER.alternateuserid
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein und `alternateuserid` ist der Wert des Felds *AlternateUserId* im Objektdeskriptor.

Ein Profil, dem der Name des Warteschlangenmanagers vorangesetzt ist, steuert die Verwendung einer alternativen Benutzer-ID in diesem Warteschlangenmanager. Ein Profil mit dem Namen der Gruppe mit gemeinsamer Warteschlange als Präfix steuert die Verwendung einer alternativen Benutzer-ID auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange. Diese alternative Benutzer-ID kann auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange von einem Benutzer mit der korrekten Zugriffsberechtigung verwendet werden. Dieser Zugriff kann auf einem einzelnen WS-Manager überschrieben werden, indem ein WS-Managerebenenprofil für diese alternative Benutzer-ID in diesem WS-Manager definiert wird.

Wenn Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit auf der Ebene des Warteschlangenmanagers als auch die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, sucht IBM MQ zuerst nach einem Profil, das dem Namen des Warteschlangenmanagers vorangestellt ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist.

Die folgende Tabelle zeigt den Zugriff bei der Angabe einer alternativen Benutzeroption.

Tabelle 46. Zugriffsebenen für alternative Benutzersicherheit	
Option 'MQOPEN', 'MQSUB' oder 'MQPUT1'	**RACF-Zugriffsebene erforderlich**
MQOO_ALTERNATE_USER_AUTHORITY MQSO_ALTERNATE_USER_AUTHORITY MQPMO_ALTERNATE_USER_AUTHORITY	UPDATE

Zusätzlich zu alternativen Sicherheitsüberprüfungen für Benutzer können auch andere Sicherheitsprüfungen für Warteschlangen, Prozesse, Namenslisten und Kontextsicherheit durchgeführt werden. Die alternative Benutzer-ID, falls angegeben, wird nur für Sicherheitsprüfungen in Warteschlangen, Prozessdefinitionen oder Namenslistenressourcen verwendet. Für alternative Benutzer-und Kontextsicherheitsüberprüfungen wird die Benutzer-ID, die die Prüfung anfordert, verwendet. Einzelheiten zum Umgang mit Benutzer-IDs finden Sie unter „Benutzer-IDs für die Sicherheitsprüfung unter z/OS” auf Seite 255. Eine Zusammenfassung der Optionen zum Öffnen und der erforderlichen Sicherheitsprüfungen bei gleichzeitig aktivierter Sicherheit der Warteschlangen, des Kontexts und der alternativen Benutzer-IDs finden Sie in Tabelle 36 auf Seite 223.

Ein alternatives Benutzerprofil gibt der anfordernden Benutzer-ID Zugriff auf Ressourcen, die der in der alternativen Benutzer-ID angegebenen Benutzer-ID zugeordnet sind. Der Lohnbuchhaltungsserver, der unter der Benutzer-ID PAYSERV auf WS-Manager QMPY ausgeführt wird, verarbeitet beispielsweise Anforderungen von Personalbenutzer-IDs, die alle mit PS beginnen. Damit die vom Lohnbuchhaltungsserver ausgeführte Arbeit unter der Benutzer-ID des anfordernden Benutzers ausgeführt werden kann, wird die alternative Benutzerberechtigung verwendet. Der Lohnbuchhaltungsserver weiß, welche Benutzer-ID als alternative Benutzer-ID angegeben werden soll, da die anfordernden Programme Nachrichten mit der Option MQPMO_DEFAULT_CONTEXT put message generieren. Im Abschnitt „Benutzer-IDs für die Sicherheitsprüfung unter z/OS” auf Seite 255 finden Sie weitere Informationen zur Position alternativer Benutzer-IDs.

Mit den folgenden RACF-Beispieldefinitionen wird das Serverprogramm zur Angabe alternativer Benutzer-IDs aktiviert, die mit der Zeichenfolge PS beginnen:

```
RDEFINE MQADMIN QMPY.ALTERNATE.USER.PS* UACC(NONE)
PERMIT QMPY.ALTERNATE.USER.PS* CLASS(MQADMIN) ID(PAYSERV) ACCESS(UPDATE)
```

Anmerkung:

1. Die *AlternateUserId* -Felder im Objektdeskriptor und im Subskriptionsdeskriptor sind 12 Byte lang. Alle 12 Byte werden in den Profilprüfungen verwendet, aber nur die ersten 8 Byte werden von IBM MQ als Benutzer-IDs verwendet. Wenn diese Benutzer-ID nicht abgeschnitten werden sollte, müssen Anwendungsprogramme, die die Anforderung machen, jede alternative Benutzer-ID über 8 Byte in etwas passendes zu übersetzen.
2. Wenn Sie MQOO_ALTERNATE_USER_AUTHORITY, MQSO_ALTERNATE_USER_AUTHORITY oder MQPMO_ALTERNATE_USER_AUTHORITY angeben und kein Feld *AlternateUserId* im Objektdeskriptor angegeben wird, wird eine Benutzer-ID mit Leerzeichen verwendet. Zum Zweck der Sicherheit der alternativen Benutzer-IDs ist die für das *AlternateUserId*-Qualifikationsmerkmal verwendete Benutzer-ID "-BLANK-". Beispiel: `RDEF MQADMIN hlq.ALTERNATE.USER.-BLANK-`.

   Wenn der Benutzer berechtigt ist, auf dieses Profil zuzugreifen, werden alle weiteren Prüfungen mit einer Benutzer-ID von Leerzeichen durchgeführt. Einzelheiten zu leeren Benutzer-IDs finden Sie unter „Leere Benutzer-IDs und UACC-Stufen” auf Seite 264.

Die Verwaltung alternativer Benutzer-IDs ist einfacher, wenn Sie über eine Namenskonvention für Benutzer-IDs verfügen, die es Ihnen ermöglicht, generische alternative Benutzerprofile zu verwenden. Wenn dies nicht der Fall ist, können Sie die RACF-Funktion RACVARS verwenden. Weitere Informationen zur Verwendung von RACVARS finden Sie im Handbuch *z/OS SecureWay Security Server RACF Security Administrator's Guide*.

Wenn eine Nachricht in eine Warteschlange gestellt wird, die mit der alternativen Benutzerberechtigung geöffnet wurde und der Kontext der Nachricht vom Warteschlangenmanager generiert wurde, wird das Feld MQMD_USER_IDENTIFIER auf die alternative Benutzer-ID gesetzt.

z/OS *Profile für Kontextsicherheit*

Wenn die Kontextsicherheit aktiv ist, müssen Sie Profile in den entsprechenden Klassen definieren und die erforderlichen Gruppen oder Benutzer-IDs auf diese Profile zugreifen lassen, um den Zugriff auf die Nachrichtenkontextinformationen zu steuern. Der Nachrichtenkontext ist innerhalb des Nachrichtendeskriptors (MQMD) enthalten.

Profile für die Kontextsicherheit verwenden

Wenn die Kontextsicherheit aktiv ist, müssen Sie ein Profil in einer der folgenden Klassen definieren, damit Benutzer auf Kontextinformationen für Nachrichten in einer bestimmten Warteschlange oder bei der Veröffentlichung in einem bestimmten Topic zugreifen können:

- Die Klasse `MQADMIN` bei Verwendung von Profilen in Großbuchstaben.
- Die Klasse `MXADMIN` bei Verwendung von Profilen in gemischter Groß-/Kleinschreibung.

Profile für die Kontextsicherheit können auf Subsystemebene oder auf Ebene der Gruppe mit gemeinsamer Warteschlange angegeben werden und das folgende Format annehmen:

```
hlq.CONTEXT.queuename
hlq.CONTEXT.topicname
```

Dabei kann *hlq* entweder der Warteschlangenmanagername oder der Name der Gruppe mit gemeinsamer Warteschlange und *queuename* und *topicname* entweder der vollständige oder generische Name der Warteschlange oder des Themas sein, für die bzw. das Sie das Kontextprofil definieren möchten.

Ein Profil mit dem Warteschlangenmanager-Namen als Präfix und ** als Warteschlangen-oder Themenname ermöglicht die Steuerung der Kontextsicherheit für alle Warteschlangen und Themen, die zu diesem Warteschlangenmanager gehören. Dies kann in einer einzelnen Warteschlange oder einem einzelnen Thema überschrieben werden, indem ein bestimmtes Profil für den Kontext in dieser Warteschlange oder diesem Thema definiert wird.

Ein Profil mit dem Namen der Gruppe mit gemeinsamer Warteschlange als Präfix und ** als Name der Warteschlange oder des Themas ermöglicht die Steuerung des Kontextes für alle Warteschlangen und Themen, die zu den Warteschlangenmanagern innerhalb der Gruppe mit gemeinsamer Warteschlange gehören. Diese kann auf einem einzelnen WS-Manager überschrieben werden, indem ein Profil auf WS-Managerebene für den Kontext auf diesem WS-Manager definiert wird, indem ein Profil angegeben wird, das dem Namen des WS-Managers vorangesetzt ist. Sie kann auch für einzelne Warteschlangen oder Themen überschrieben werden, indem ein Profil mit dem Namen der Warteschlange oder des Themas als Suffix angegeben wird.

Wenn Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit auf der Ebene des Warteschlangenmanagers als auch die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, sucht IBM MQ zuerst nach einem Profil, das dem Namen des Warteschlangenmanagers vorangestellt ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist.

Sie müssen erlauben, dass die erforderlichen Gruppen oder Benutzer-IDs auf dieses Profil zugreifen. Die folgende Tabelle zeigt die erforderliche Zugriffsebene, abhängig von der Spezifikation der Kontextoptionen, wenn die Warteschlange geöffnet wird.

Tabelle 47. Zugriffsebenen für die Kontextsicherheit

MQOPEN-oder MQPUT1-Option	**RACF Zugriffsebene erforderlich für hlq.CONTEXT.queuename oder hlq.CONTEXT.topicname**
MQPMO_NO_CONTEXT	Keine Kontextsicherheitsüberprüfung

Tabelle 47. Zugriffsebenen für die Kontextsicherheit (Forts.)	
MQOPEN-oder MQPUT1-Option	**RACF Zugriffsebene erforderlich für hlq.CONTEXT.queuename oder hlq.CONTEXT.topicname**
MQPMO_DEFAULT_CONTEXT	Keine Kontextsicherheitsüberprüfung
MQOO_SAVE_ALL_CONTEXT	Keine Kontextsicherheitsüberprüfung
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	READ
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	READ
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	CONTROL
MQOO_OUTPUT oder MQPUT1(USAGE (XMITQ))	CONTROL
MQSUB-Option	
MQSO_SET_IDENTITY_CONTEXT**(Anmerkung 2)**	UPDATE

Anmerkung:

1. Die Benutzer-IDs, die für die verteilte Steuerung von Warteschlangen verwendet werden, benötigen CONTROL-Zugriff auf `hlq.CONTEXT.queuename` , um Nachrichten in die Zielwarteschlange zu stellen. Unter „Vom Kanalinitiator verwendete Benutzer-IDs” auf Seite 259 finden Sie Informationen zu den verwendeten Benutzer-IDs.
2. Wenn Sie in der MQSUB-Anforderung mit den Optionen MQSO_CREATE oder MQSO_ALTER angeben, dass Sie die Identitätskontextfelder in der MQSD-Struktur angeben möchten, müssen Sie die Option MQSO_SET_IDENTITY_CONTEXT angeben. Sie benötigen außerdem die entsprechende Berechtigung für das Kontextprofil für die Zielwarteschlange.

Wenn Sie Befehle in die Eingabewarteschlange des Systembefehls eingeben, verwenden Sie die Standardkontextnachrichtenoption, um die richtige Benutzer-ID dem Befehl zuzuordnen.

Beispielsweise kann das von IBM MQ bereitgestellte Dienstprogramm CSQUTIL zum Auslagern und erneuten Laden von Nachrichten in Warteschlangen verwendet werden. Wenn ausgelagerte Nachrichten in eine Warteschlange zurückgespeichert werden, verwendet das Dienstprogramm CSQUTIL die Option MQOO_SET_ALL_CONTEXT, um die Nachrichten in ihren ursprünglichen Zustand zurückzugeben. Zusätzlich zur Warteschlangensicherheit, die für diese offene Option erforderlich ist, ist auch die Kontextberechtigung erforderlich. Wenn diese Berechtigung beispielsweise für die Gruppe BACKGRP auf WS-Manager MQS1 erforderlich ist, würde dies wie folgt definiert werden:

```
RDEFINE MQADMIN MQS1.CONTEXT.** UACC(NONE)
PERMIT MQS1.CONTEXT.** CLASS(MQADMIN) ID(BACKGRP) ACCESS(CONTROL)
```

Abhängig von den angegebenen Optionen und den Typen der ausgeführten Sicherheit können auch andere Arten von Sicherheitsprüfungen beim Öffnen der Warteschlange auftreten. Dies umfasst die Sicherheit für Warteschlangen (siehe „Profile für die Warteschlangensicherheit” auf Seite 214) und die Sicherheit für alternative Benutzer-IDs (siehe „Profile für alternative Benutzersicherheit” auf Seite 231). Eine Zusammenfassung der Optionen zum Öffnen und der erforderlichen Sicherheitsprüfungen bei gleichzeitig aktivierter Sicherheit der Warteschlangen, des Kontexts und der alternativen Benutzer-IDs finden Sie in Tabelle 36 auf Seite 223.

Kontextsicherheit der Systemwarteschlange

Auf viele der Systemwarteschlangen wird durch die Nebenkomponenten von IBM MQ zugegriffen, z. B. auf den Adressraum des Kanalinitiators, und den von der IBM MQ Console und der REST API verwendeten mqweb-Server.

Die Benutzer-IDs, unter denen diese ausgeführt werden, müssen RACF-Zugriff auf diese Warteschlangen erhalten, wie in Tabelle 48 auf Seite 235 dargestellt.

Tabelle 48. Zugriff auf die SYSTEM-Warteschlangen für Kontextoperationen erforderlich

SYSTEM-Warteschlange	Kanalinitiator für verteilte Steuerung von Warteschlangen	mqweb-Server
SYSTEM.ADMIN.COMMAND.QUEUE	-	CONTROL
SYSTEM.BROKER.CONTROL.QUEUE	CONTROL	-
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	CONTROL	-
SYSTEM.CHANNEL.SYNCQ	CONTROL	-
SYSTEM.CLUSTER.COMMAND.QUEUE	CONTROL	-
SYSTEM.CLUSTER.TRANSMIT.QUEUE	CONTROL	-

z/OS *Profile für die Befehlssicherheit*

Um die Sicherheitsprüfung für Befehle zu aktivieren, fügen Sie Profile zur MQCMDS-Klasse hinzu. Die Profilnamen basieren auf den MQSC-Befehlen, steuern aber sowohl die MQSC-als auch die PCF-Befehle. Profile können für einen Warteschlangenmanager oder eine Gruppe mit gemeinsamer Warteschlange angewendet werden.

Wenn Sie die Sicherheitsprüfung für Befehle wünschen (so dass Sie das Befehlssicherheitsschalterprofil hlq.NO.CMD.CHECKS nicht definiert haben), müssen Sie Profile zur MQCMDS-Klasse hinzufügen.

Dieselben Sicherheitsprofile steuern sowohl die MQSC-als auch die PCF-Befehle. Die Namen der RACF-Profile für die Überprüfung der Befehlssicherheit basieren auf den MQSC-Befehlsnamen selbst. Diese Profile haben das folgende Format:

```
hlq.verb.pkw
```

Dabei kann `hlq` entweder `qmgr-name` (Name des Warteschlangenmanagers) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein, `verb` ist der Verb-Teil des Befehlsnamens, z. B. ALTER, und `pkw` ist der Objekttyp, z. B. QLOCAL für eine lokale Warteschlange.

Daher lautet der Profilname für den Befehl ALTER QLOCAL im Subsystem CSQ1:

```
CSQ1.ALTER.QLOCAL
```

Sie können generische Profile verwenden, um Gruppen von Befehlen zu schützen, so dass weniger Profile verwaltet werden müssen und daher weniger Zugriffslisten vorhanden sind. Ziehen Sie die Erstellung eines generischen Profils in Betracht, das für alle Befehle gilt, die nicht durch ein spezifiseres Profil geschützt sind. Definieren Sie dieses Profil mit UACC(NONE) und erteilen Sie den ALTER-Zugriff nur den RACF-Gruppen, die Administratoren enthalten. Anschließend können Sie ein generisches Profil erstellen, das auf alle DISPLAY-Befehle anwendbar ist und weit verbreiteten Zugriff auf diese Befehle erteilt. Innerhalb dieser beiden Extreme können Sie Gruppen von Benutzern identifizieren, die Zugriff auf bestimmte Befehlsgruppen benötigen. In diesem Fall können Sie Profile für diese Gruppen erstellen und den RACF-Gruppen der betreffenden Benutzerklasse Zugriff gewähren. Vermeiden Sie es, Benutzern Zugriff auf Befehle zu erteilen, die sie nicht benötigen: Wenden Sie das Prinzip der geringsten Berechtigung an, damit die Benutzer nur Zugriff auf die Befehle haben, die für ihre Jobs erforderlich sind.

Ein Profil, das dem Namen des Warteschlangenmanagers vorangesetzt ist, steuert die Verwendung des Befehls in diesem Warteschlangenmanager. Ein Profil, das durch den Namen der Gruppe mit gemeinsamer Warteschlange festgelegt wird, steuert die Verwendung des Befehls auf allen Warteschlangenmanagern innerhalb der Gruppe mit gemeinsamer Warteschlange. Dieser Zugriff kann auf einem einzelnen WS-Manager außer Kraft gesetzt werden, indem ein WS-Managerebenenprofil für diesen Befehl in diesem Warteschlangenmanager definiert wird.

Wenn Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit auf der Ebene des Warteschlangenmanagers als auch die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, sucht IBM MQ nach einem Profil, das dem Namen des Warteschlangenmanagers vorangestellt ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist.

Durch die Einrichtung von Befehlsprofilen auf WS-Managerebene kann ein Benutzer von der Ausgabe von Befehlen auf einem bestimmten Warteschlangenmanager eingeschränkt werden. Alternativ können Sie für jedes Befehlsverb ein Profil für eine Gruppe mit gemeinsamer Warteschlange definieren, und alle Sicherheitsprüfungen werden für dieses Profil anstelle von einzelnen Warteschlangenmanagern ausgeführt.

Wenn sowohl die Sicherheit der Subsystemsicherheit als auch die Sicherheit der Gruppe mit gemeinsamer Warteschlange aktiv sind und kein lokales Profil gefunden wird, wird eine Befehlssicherheitsüberprüfung ausgeführt, um zu prüfen, ob der Benutzer Zugriff auf ein Profil für die Gruppe mit gemeinsamer Warteschlange hat.

Wenn Sie das Attribut CMDSCOPE verwenden, um einen Befehl an andere Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange weiterzuleiten, wird die Sicherheit auf jedem Warteschlangenmanager überprüft, auf dem der Befehl ausgeführt wird, aber nicht unbedingt auf dem Warteschlangenmanager, auf dem der Befehl eingegeben wird.

Tabelle 49 auf Seite 236 zeigt für jeden IBM MQ-MQSC-Befehl die Profile, die für die Befehlssicherheitsprüfung erforderlich sind, und die entsprechende Zugriffsebene für jedes Profil in der MQCMDS-Klasse, an.

Tabelle 50 auf Seite 242 zeigt für jeden IBM MQ-PCF-Befehl die für die Ausführung der Befehlssicherheitsprüfung erforderlichen Profile sowie die zugehörige Zugriffsebene für jedes Profil in der MQCMDS-Klasse.

Tabelle 49. MQSC-Befehle, -Profile und deren Zugriffsebenen

Befehl	**Befehlsprofil für MQCMDS**	**Zugriffsebene für MQCMDS**	**Befehlsressourcenprofil für MQADMIN oder MXADMIN**	**Zugriffsebene für MQADMIN oder MXADMIN**
ALTER AUTHINFO	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
ALTER BUFFPOOL	hlq.ALTER.BUFFPOOL	ALTER	Keine Prüfung	-
ALTER CFSTRUCT	hlq.ALTER.CFSTRUCT	ALTER	Keine Prüfung	-
ALTER CHANNEL	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
ALTER NAMELIST	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
ALTER PROCESS	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
ALTER PSID	hlq.ALTER.PSID	ALTER	Keine Prüfung	-
ALTER QALIAS	hlq.ALTER.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
ALTER QLOCAL	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	Keine Prüfung	-

Tabelle 49. MQSC-Befehle, -Profile und deren Zugriffsebenen (Forts.)

Befehl	Befehlsprofil für MQCMDS	Zugriffs-ebene für MQCMDS	Befehlsressourcenprofil für MQADMIN oder MXADMIN	Zugriffs-ebene für MQADMIN oder MXAD-MIN
ALTER QMODEL	hlq.ALTER.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QREMOTE	hlq.ALTER.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
ALTER SECURITY	hlq.ALTER.SECURITY	ALTER	Keine Prüfung	-
ALTER SMDS	hlq.ALTER.SMDS	ALTER	Keine Prüfung	-
ALTER STGCLASS	hlq.ALTER.STGCLASS	ALTER	Keine Prüfung	-
ALTER SUB	hlq.ALTER.SUB	ALTER	Keine Prüfung	-
ALTER TOPIC	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
ALTER TRACE	hlq.ALTER.TRACE	ALTER	Keine Prüfung	-
ARCHIVE LOG	hlq.ARCHIVE.LOG	STEUE-RUNG	Keine Prüfung	-
BACKUP CFSTRUCT	hlq.BACKUP.CFSTRUCT	STEUE-RUNG	Keine Prüfung	-
CLEAR QLOCAL	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
CLEAR TOPICSTR „3” auf Seite 241	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
AUTHINFO DEFINIER	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcena-me	ALTER
DEFINE BUFFPOOL	hlq.DEFINE.BUFFPOOL	ALTER	Keine Prüfung	-
CFSTRUCT DEFINE CFSTRU	hlq.DEFINE.CFSTRUCT	ALTER	Keine Prüfung	-
CHANNEL DEFINE CHANNEL	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
PROTOKOLL	hlq.DEFINE.LOG	ALTER	Keine Prüfung	-
DEFINE MAXSMSGS	hlq.DEFINE.MAXSMSGS	ALTER	Keine Prüfung	-
NAMELIST DEFINI	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
PROZESS DEFI	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DEFINE PSID	hlq.DEFINE.PSID	ALTER	Keine Prüfung	-
DEFINE QALIAS	hlq.DEFINE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
QLOCAL DEFINIER	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
QMODEL DEFINIER	hlq.DEFINE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
QREMOTE DEFINIER	hlq.DEFINE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
STGKLASSE DEFINI	hlq.DEFINE.STGCLASS	ALTER	Keine Prüfung	-
SUB DEFINI	hlq.DEFINE.SUB	ALTER	Keine Prüfung	-
TOPIC DEFINI	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER

Tabelle 49. MQSC-Befehle, -Profile und deren Zugriffsebenen (Forts.)

Befehl	Befehlsprofil für MQCMDS	Zugriffsebene für MQCMDS	Befehlsressourcenprofil für MQADMIN oder MXADMIN	Zugriffsebene für MQADMIN oder MXADMIN
DELETE AUTHINFO	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DELETE BUFFPOOL	hlq.DELETE.BUFFPOOL	ALTER	Keine Prüfung	-
DELETE CFSTRUCT	hlq.DELETE.CFSTRUCT	ALTER	Keine Prüfung	-
DELETE CHANNEL	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DELETE NAMELIST	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
DELETE PROCESS	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DELETE PSID	hlq.DELETE.PSID	ALTER	Keine Prüfung	-
DELETE QALIAS	hlq.DELETE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DELETE QLOCAL	hlq.DELETE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QMODEL	hlq.DELETE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QREMOTE	hlq.DELETE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DELETE STGCLASS	hlq.DELETE.STGCLASS	ALTER	Keine Prüfung	-
DELETE SUB	hlq.DELETE.SUB	ALTER	Keine Prüfung	-
DELETE TOPIC	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DISPLAY ARCHIVE „1” auf Seite 241	hlq.DISPLAY.ARCHIVE	READ	Keine Prüfung	-
DISPLAY AUTHINFO	hlq.DISPLAY.AUTHINFO	READ	Keine Prüfung	-
DISPLAY CFSTATUS	hlq.DISPLAY.CFSTATUS	READ	Keine Prüfung	-
ANZEIGEN CFSTRUCT	hlq.DISPLAY.CFSTRUCT	READ	Keine Prüfung	-
ANZEIGEN CHANNEL	hlq.DISPLAY.CHANNEL	READ	Keine Prüfung	-
ANZEIGEN CHINIT	hlq.DISPLAY.CHINIT	READ	Keine Prüfung	-
ANZEIGEN CHLAUTH	hlq.DISPLAY.CHLAUTH	READ	Keine Prüfung	-
ANZEIGEN CHSTATUS	hlq.DISPLAY.CHSTATUS	READ	Keine Prüfung	-
DISPLAY CLUSQMGR	hlq.DISPLAY.CLUSQMGR	READ	Keine Prüfung	-
ANZEIGEN CMDSERV	hlq.DISPLAY.CMDSERV	READ	Keine Prüfung	-
DISPLAY CONN „1” auf Seite 241	hlq.DISPLAY.CONN	READ	Keine Prüfung	-
ANZEIGEN G	hlq.DISPLAY.GROUP	READ	Keine Prüfung	-
DISPLAY LOG „1” auf Seite 241	hlq.DISPLAY.LOG	READ	Keine Prüfung	-
ANZEIGEN MAXSMSGS	hlq.DISPLAY.MAXSMSGS	READ	Keine Prüfung	-

Tabelle 49. MQSC-Befehle, -Profile und deren Zugriffsebenen (Forts.)

Befehl	Befehlsprofil für MQCMDS	Zugriffsebene für MQCMDS	Befehlsressourcenprofil für MQADMIN oder MXADMIN	Zugriffsebene für MQADMIN oder MXADMIN
ANZEIGEN NAMELIST	hlq.DISPLAY.NAMELIST	READ	Keine Prüfung	-
ANZEIGEN PROZ	hlq.DISPLAY.PROCESS	READ	Keine Prüfung	-
DISPLAY PUBSUB	hlq.DISPLAY.PUBSUB	READ	Keine Prüfung	-
DISPLAY QALIAS	hlq.DISPLAY.QALIAS	READ	Keine Prüfung	-
ANZEIGEN QCLUSTER	hlq.DISPLAY.QCLUSTER	READ	Keine Prüfung	-
ANZEIGEN QLOCAL	hlq.DISPLAY.QLOCAL	READ	Keine Prüfung	-
ANZEIGEN QMGR	hlq.DISPLAY.QMGR	READ	Keine Prüfung	-
ANZEIGEN QMODEL	hlq.DISPLAY.QMODEL	READ	Keine Prüfung	-
ANZEIGEN QREMOTE	hlq.DISPLAY.QREMOTE	READ	Keine Prüfung	-
ANZEIGEN QSTATUS	hlq.DISPLAY.QSTATUS	READ	Keine Prüfung	-
ANZEIGEN QUEUE	hlq.DISPLAY.QUEUE	READ	Keine Prüfung	-
ANZEIGEN SBSTATUS	hlq.DISPLAY.SBSTATUS	READ	Keine Prüfung	-
ANZEIGEN SMDS	hlq.DISPLAY.SMDS	READ	Keine Prüfung	-
ANZEIGEN SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	Keine Prüfung	-
ANZEIGEN SUB	hlq.DISPLAY.SUB	READ	Keine Prüfung	-
DISPLAY SECURITY	hlq.DISPLAY.SECURITY	READ	Keine Prüfung	-
ANZEIGEN STGKLASSE	hlq.DISPLAY.STGCLASS	READ	Keine Prüfung	-
DISPLAY SYSTEM „1" auf Seite 241	hlq.DISPLAY.SYSTEM	READ	Keine Prüfung	-
ANZEIGEN THREAD	hlq.DISPLAY.THREAD	READ	Keine Prüfung	-
ANZEIGEN TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	Keine Prüfung	-
ANZEIGEN TOPIC	hlq.DISPLAY.TOPIC	READ	Keine Prüfung	-
ANZEIGEN TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	Keine Prüfung	-
ANZEIGEN TRACE	hlq.DISPLAY.TRACE	READ	Keine Prüfung	-
DISPLAY USAGE „1" auf Seite 241	hlq.DISPLAY.USAGE	READ	Keine Prüfung	-
MOVE QLOCAL	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
PING CHANNEL	hlq.PING.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
RECOVER BSDS	hlq.RECOVER.BSDS	STEUERUNG	Keine Prüfung	-

Tabelle 49. MQSC-Befehle, -Profile und deren Zugriffsebenen (Forts.)

Befehl	Befehlsprofil für MQCMDS	Zugriffsebene für MQCMDS	Befehlsressourcenprofil für MQADMIN oder MXADMIN	Zugriffsebene für MQADMIN oder MXADMIN
RECOVER CFSTRUCT	hlq.RECOVER.CFSTRUCT	STEUERUNG	Keine Prüfung	-
REFRESH CLUSTER	hlq.REFRESH.CLUSTER	ALTER	Keine Prüfung	-
REFRESH QMGR	hlq.REFRESH.QMGR	ALTER	Keine Prüfung	-
REFRESH SECURITY	hlq.REFRESH.SECURITY	ALTER	Keine Prüfung	-
RESET CFSTRUCT	hlq.RESET.CFSTRUCT	STEUERUNG	Keine Prüfung	-
RESET CHANNEL	hlq.RESET.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
RESET CLUSTER	hlq.RESET.CLUSTER	STEUERUNG	Keine Prüfung	-
RESET QMGR	hlq.RESET.QMGR	STEUERUNG	Keine Prüfung	-
RESET QSTATS	hlq.RESET.QSTATS	STEUERUNG	hlq.QUEUE.queue	STEUERUNG
RESET SMDS	hlq.RESET.SMDS	STEUERUNG	Keine Prüfung	-
RESET TPIPE	hlq.RESET.TPIPE	STEUERUNG	Keine Prüfung	-
GELÖST-CHANNEL	hlq.RESOLVE.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
INDOUBT GELÖST	hlq.RESOLVE.INDOUBT	STEUERUNG	Keine Prüfung	-
RESUME QMGR	hlq.RESUME.QMGR	STEUERUNG	Keine Prüfung	-
SICHERHEIT ÜBERPRÜFEN	hlq.RVERIFY.SECURITY	ALTER	Keine Prüfung	-
SET ARCHIVE	hlq.SET.ARCHIVE	STEUERUNG	Keine Prüfung	-
SET CHLAUTH	hlq.SET.CHLAUTH	STEUERUNG	Keine Prüfung	-
SET LOG	hlq.SET.LOG	STEUERUNG	Keine Prüfung	-
SET SYSTEM	hlq.SET.SYSTEM	STEUERUNG	Keine Prüfung	-
START CHANNEL	hlq.START.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG

Tabelle 49. MQSC-Befehle, -Profile und deren Zugriffsebenen (Forts.)

Befehl	Befehlsprofil für MQCMDS	Zugriffsebene für MQCMDS	Befehlsressourcenprofil für MQADMIN oder MXADMIN	Zugriffsebene für MQADMIN oder MXADMIN
START CHINIT „4” auf Seite 242	hlq.START.CHINIT	STEUERUNG	Keine Prüfung	-
START CMDSERV	hlq.START.CMDSERV	STEUERUNG	Keine Prüfung	-
START LISTENER	hlq.START.LISTENER	STEUERUNG	Keine Prüfung	-
START QMGR	Ohne „2” auf Seite 241	-	-	-
START SMDSCONN	hlq.START.SMDSCONN	STEUERUNG	Keine Prüfung	-
START TRACE	hlq.START.TRACE	STEUERUNG	Keine Prüfung	-
STOP CHANNEL	hlq.STOP.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
STOP CHINIT	hlq.STOP.CHINIT	STEUERUNG	Keine Prüfung	-
STOP CMDSERV	hlq.STOP.CMDSERV	STEUERUNG	Keine Prüfung	-
STOP LISTENER	hlq.STOP.LISTENER	STEUERUNG	Keine Prüfung	-
STOP QMGR	hlq.STOP.QMGR	STEUERUNG	Keine Prüfung	-
STOP SMDSCONN	hlq.STOP.SMDSCONN	STEUERUNG	Keine Prüfung	-
STOP-TRACE	hlq.STOP.TRACE	STEUERUNG	Keine Prüfung	-
SUSPEND QMGR	hlq.SUSPEND.QMGR	STEUERUNG	Keine Prüfung	-

Anmerkungen:

1. Diese Befehle werden möglicherweise intern vom WS-Manager ausgegeben; in diesen Fällen wird keine Berechtigung geprüft.
2. IBM MQ überprüft nicht des Benutzers, der den Befehl START QMGR ausgibt. Sie können jedoch RACF oder eine alternative Sicherheitsfunktion verwenden, um den Zugriff auf den Befehl START xxxxMSTR zu steuern, der als Ergebnis des Befehls START QMGR ausgegeben wurde. Dies wird durch die Steuerung des Zugriffs auf das Profil MVS.START.STC.xxxxMSTR in der Klasse für RACF-Operatorbefehle (OPERCMDS) erreicht. Einzelheiten zu diesem Vorgehen finden Sie im Handbuch *z/OS SecureWay Security Server RACF Security Administrator's Guide*. Wenn Sie dieses Verfahren verwenden und ein nicht berechtigter Benutzer versucht, den WS-Manager zu starten, wird er mit dem Ursachencode 00F30216 beendet.
3. Die **hlq.TOPIC.topic** -Ressource bezieht sich auf das Themenobjekt, das von TOPICSTR abgeleitet wurde. Weitere Informationen finden Sie unter „Publish/Subscribe-Sicherheit” auf Seite 517

4. Bei Releases vor IBM MQ for z/OS V6 wurde die Sicherheitsprüfung für MVS.START.STC.CSQ1CHIN durchgeführt. In IBM MQ for z/OS V6 und höher ist an den Ressourcenname das zusätzliche Qualifikationsmerkmal JOBNAME angehängt. Dies kann zu Problemen beim Starten des Kanalinitiators führen.

   Um das Problem zu beheben, ersetzen Sie MVS.START.STC. *ssid* CHIN mit einem Profil für eine Ressource mit dem Namen MVS.START.STC *ssid* CHIN **.*** oder MVS.START.STC. *ssid* CHIN. *ssid* CHIN, wobei *ssid* für die Subsystem-ID des Warteschlangenmanagers steht. Hierfür ist die RACF-Berechtigung UPDATE erforderlich. Weitere Informationen finden Sie in der Produktdokumentation zu z/OS unter *Operation planning, MVS Commands, RACF Access Authorities, and Resource Names*.

   Der Parameter START für *ssid* MSTR enthält den Parameter JOBNAME= nicht. Aus Gründen der Konsistenz kann es sein, dass Sie das Profil für MVS.START.STC.ssidMSTR auf MVS.START.STC.ssidMSTR. * aktualisieren.

Tabelle 50. PCF-Befehle, -Profile und deren Zugriffsebenen

Befehl	**Befehlsprofil für MQCMDS**	**Zugriffsebene für MQCMDS**	**Befehlsressourcenprofil für MQADMIN oder MXADMIN**	**Zugriffsebene für MQADMIN oder MXADMIN**
CF-Struktur sichern	hlq.BACKUP.CFSTRUCT	STEUERUNG	Keine Prüfung	-
Authentifizierungsinformationsobjekt ändern	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
CF-Struktur ändern	hlq.ALTER.CFSTRUCT	ALTER	Keine Prüfung	-
Kanal ändern	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Namensliste ändern	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Prozess ändern	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Warteschlange ändern	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Warteschlangenmanager ändern	hlq.ALTER.QMGR	ALTER	Keine Prüfung	-
Sicherheit ändern	hlq.ALTER.SECURITY	ALTER	Keine Prüfung	-
SMDS ändern	hlq.ALTER.SMDS	ALTER	Keine Prüfung	-
Speicherklasse ändern	hlq.ALTER.STGCLASS	ALTER	Keine Prüfung	-
Subskription ändern	hlq.ALTER.SUB	ALTER	Keine Prüfung	-
Thema ändern	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Warteschlange löschen	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Clear Topic String (Themenzeichenfolge löschen)„1” auf Seite 246	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
Authentifizierungsinformationsobjekt kopieren	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
CF-Struktur kopieren	hlq.DEFINE.CFSTRUCT	ALTER	Keine Prüfung	-
Kanal kopieren	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Namensliste kopieren	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Prozess kopieren	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Warteschlange kopieren	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Subskription kopieren	hlq.DEFINE.SUB	ALTER	Keine Prüfung	-
Speicherklasse kopieren	hlq.DEFINE.STGCLASS	ALTER	Keine Prüfung	-

Tabelle 50. PCF-Befehle, -Profile und deren Zugriffsebenen (Forts.)

Befehl	Befehlsprofil für MQCMDS	Zugriffsebene für MQCMDS	Befehlsressourcenprofil für MQADMIN oder MXADMIN	Zugriffsebene für MQADMIN oder MXADMIN
Thema kopieren	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Authentifizierungsinformationsobjekt erstellen	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
CF-Struktur erstellen	hlq.DEFINE.CFSTRUCT	ALTER	Keine Prüfung	-
Kanal erstellen	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Namensliste erstellen	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Prozess erstellen	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Warteschlange erstellen	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Speicherklasse erstellen	hlq.DEFINE.STGCLASS	ALTER	Keine Prüfung	-
Subskription erstellen	hlq.DEFINE.SUB	ALTER	Keine Prüfung	-
Thema erstellen	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
CF-Struktur löschen	hlq.DELETE.CFSTRUCT	ALTER	Keine Prüfung	-
Kanal löschen	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Namelist	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Prozess löschen	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Warteschlange löschen	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Speicherklasse löschen	hlq.DELETE.STGCLASS	ALTER	Keine Prüfung	-
Subskription löschen	hlq.DELETE.SUB	ALTER	Keine Prüfung	-
Thema löschen	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Archiv erfragen	hlq.DISPLAY.ARCHIVE	READ	Keine Prüfung	-
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	Keine Prüfung	-
Objektnamen für Authentifizierungsinformationen anfragen	hlq.DISPLAY.AUTHINFO	READ	Keine Prüfung	-
CF-Struktur erfragen	hlq.DISPLAY.CFSTRUCT	READ	Keine Prüfung	-
CF-Strukturnamen anfragen	hlq.DISPLAY.CFSTRUCT	READ	Keine Prüfung	-
CF-Strukturstatus abgefragt	hlq.DISPLAY.CFSTATUS	READ	Keine Prüfung	-
Kanalinquire	hlq.DISPLAY.CHANNEL	READ	Keine Prüfung	-
Kanalauthentifizierungsdatensätze abgefragt	hlq.DISPLAY.CHLAUTH	READ	Keine Prüfung	-
Kanalinitiator inquire	hlq.DISPLAY.CHINIT	READ	Keine Prüfung	-
Kanalnamen inquire	hlq.DISPLAY.CHANNEL	READ	Keine Prüfung	-

Tabelle 50. PCF-Befehle, -Profile und deren Zugriffsebenen (Forts.)

Befehl	Befehlsprofil für MQCMDS	Zugriffsebene für MQCMDS	Befehlsressourcenprofil für MQADMIN oder MXADMIN	Zugriffsebene für MQADMIN oder MXADMIN
Kanalstatus abgefragt	hlq.DISPLAY.CHSTATUS	READ	Keine Prüfung	-
Clusterwarteschlangenmanager anfragen	hlq.DISPLAY.CLUSQMGR	READ	Keine Prüfung	-
Verbindung anfragen	hlq.DISPLAY.CONNPCF	READ	Keine Prüfung	-
Inquire-Gruppe	hlq.DISPLAY.GROUP	READ	Keine Prüfung	-
Protokoll anfragen	hlq.DISPLAY.LOG	READ	Keine Prüfung	-
Namensliste inquire	hlq.DISPLAY.NAMELIST	READ	Keine Prüfung	-
Namen der Namensliste aufrufen	hlq.DISPLAY.NAMELIST	READ	Keine Prüfung	-
Prozess inquire	hlq.DISPLAY.PROCESS	READ	Keine Prüfung	-
Prozessnamen inquire	hlq.DISPLAY.PROCESS	READ	Keine Prüfung	-
Publish/Subscribe-Status inquire	hlq.DISPLAY.PUBSUB	READ	Keine Prüfung	-
Warteschlange einfragen	hlq.DISPLAY.QUEUE	READ	Keine Prüfung	-
Warteschlangenmanager abfragen	hlq.DISPLAY.QMGR	READ	Keine Prüfung	-
Warteschlangennamen inquire	hlq.DISPLAY.QUEUE	READ	Keine Prüfung	-
Warteschlangenstatus abgefragt	hlq.DISPLAY.QSTATUS	READ	Keine Prüfung	-
Sicherheit inquire	hlq.DISPLAY.SECURITY	READ	Keine Prüfung	-
SMDS anfragen	hlq.DISPLAY.SMDS	READ	Keine Prüfung	-
SMDSCONN abgefragt	hlq.DISPLAY.SMDSCONN	READ	Keine Prüfung	-
Speicherklasse inquire	hlq.DISPLAY.STGCLASS	READ	Keine Prüfung	-
Speicherklassennamen inquire	hlq.DISPLAY.STGCLASS	READ	Keine Prüfung	-
Subskription inquire	hlq.INQUIRE.SUB	READ	Keine Prüfung	-
Subskriptionsstatus abfragen	hlq.INQUIRE.SBSTATUS	READ	Keine Prüfung	-
System abgefragt	hlq.DISPLAY.SYSTEM	READ	Keine Prüfung	-
Thema erfragen	hlq.DISPLAY.TOPIC	READ	Keine Prüfung	-
Themennamen inquire	hlq.DISPLAY.TOPIC	READ	Keine Prüfung	-
Themenstatus inquire	hlq.DISPLAY.TPSTATUS	READ	Keine Prüfung	-
Belegung inquire	hlq.DISPLAY.USAGE	READ	Keine Prüfung	-
Warteschlange verschieben	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
Pingkanal	hlq.PING.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG

Tabelle 50. PCF-Befehle, -Profile und deren Zugriffsebenen (Forts.)

Befehl	Befehlsprofil für MQCMDS	Zugriffsebene für MQCMDS	Befehlsressourcenprofil für MQADMIN oder MXADMIN	Zugriffsebene für MQADMIN oder MXADMIN
CF-Struktur wiederherstellen	hlq.RECOVER.CFSTRUCT	STEUERUNG	Keine Prüfung	-
Cluster aktualisieren	hlq.REFRESH.CLUSTER	ALTER	Keine Prüfung	-
WS-Manager aktualisieren	hlq.REFRESH.QMGR	ALTER	Keine Prüfung	-
Sicherheit aktualisieren	hlq.REFRESH.SECURITY	ALTER	Keine Prüfung	-
CF-Struktur zurücksetzen	hlq.RESET.CFSTRUCT	STEUERUNG	Keine Prüfung	-
Kanal zurücksetzen	hlq.RESET.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
Cluster zurücksetzen	hlq.RESET.CLUSTER	STEUERUNG	Keine Prüfung	-
Warteschlangenmanager zurücksetzen	hlq.RESET.QMGR	STEUERUNG	Keine Prüfung	-
Warteschlangenstatistik zurücksetzen	hlq.RESET.QSTATS	STEUERUNG	hlq.QUEUE.queue	STEUERUNG
SMDS zurücksetzen	hlq.RESET.SMDS	STEUERUNG	Keine Prüfung	-
Kanal auflösen	hlq.RESOLVE.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
WS-Manager wiederaufnehmen	hlq.RESUME.QMGR	STEUERUNG	Keine Prüfung	-
WS-Manager-Cluster wiederaufnehmen	hlq.RESUME.QMGR	STEUERUNG	Keine Prüfung	-
Reverify Security	hlq.RVERIFY.SECURITY	ALTER	Keine Prüfung	-
Archiv festlegen	hlq.SET.ARCHIVE	STEUERUNG	Keine Prüfung	-
Kanalauthentifizierungsdatensatz festlegen	hlq.SET.CHLAUTH	STEUERUNG	Keine Prüfung	-
Protokoll festlegen	hlq.SET.LOG	STEUERUNG	Keine Prüfung	-
System festlegen	hlq.SET.SYSTEM	STEUERUNG	Keine Prüfung	-
Kanal starten	hlq.START.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
Kanalinitiator starten	hlq.START.CHINIT	STEUERUNG	Keine Prüfung	-
Kanal-Listener starten	hlq.START.LISTENER	STEUERUNG	Keine Prüfung	-
SMDS-Verbindung starten	hlq.START.SMDSCONN	STEUERUNG	Keine Prüfung	-
Kanal stoppen	hlq.STOP.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
Kanalinitiator stoppen	hlq.STOP.CHINIT	STEUERUNG	Keine Prüfung	-
Kanallistener stoppen	hlq.STOP.LISTENER	STEUERUNG	Keine Prüfung	-
SMDS-Verbindung stoppen	hlq.STOP.SMDSCONN	STEUERUNG	Keine Prüfung	-
WS-Manager aussetzen	hlq.SUSPEND.QMGR	STEUERUNG	Keine Prüfung	-

Tabelle 50. PCF-Befehle, -Profile und deren Zugriffsebenen (Forts.)				
Befehl	**Befehlsprofil für MQCMDS**	**Zugriffsebene für MQCMDS**	**Befehlsressourcenprofil für MQADMIN oder MXADMIN**	**Zugriffsebene für MQADMIN oder MXADMIN**
Clusterwarteschlangenmanager-Cluster aussetzen	hlq.SUSPEND.QMGR	STEUERUNG	Keine Prüfung	-

Anmerkungen:

1. Die **hlq.TOPIC.topic** -Ressource bezieht sich auf das Themenobjekt, das von TOPICSTR abgeleitet wurde. Weitere Informationen finden Sie unter „Publish/Subscribe-Sicherheit” auf Seite 517

Im Abschnitt „IBM MQ Console - erforderliche Profile für die Befehlssicherheit” auf Seite 246 finden Sie Einzelheiten zu den IBM MQ-PCF-Profilen, die bei der Verwendung der IBM MQ Console erforderlich sind.

z/OS *IBM MQ Console - erforderliche Profile für die Befehlssicherheit*

In der IBM MQ Console von einem Benutzer in der `MQWebAdmin`- oder `MQWebAdminRO`-Rolle ausgeführte Operationen finden unter dem Sicherheitskontext der Benutzer-ID der gestarteten Task 'mqweb' statt. Wenn Sie die IBM MQ Console verwenden möchten, muss die Task-Benutzer-ID für den MQWeb-Server die Berechtigung zum Absetzen bestimmter PCF-Befehle erteilen.

Tabelle 51 auf Seite 246 zeigt für jeden IBM MQ-PCF-Befehl die erforderlichen Befehlssicherheitprofile und die jeweilige Zugriffsebene für jedes Profil in der MQCMDS-Klasse, das von der IBM MQ Console benötigt wird.

Tabelle 51. PCF-Befehle, Profile und zugehörige Zugriffsebenen für die IBM MQ Console				
Befehl	**Befehlsprofil für MQCMDS**	**Zugriffsebene für MQCMDS**	**Befehlsressourcenprofil für MQADMIN oder MXADMIN**	**Zugriffsebene für MQADMIN oder MXADMIN**
Authentifizierungsinformationsobjekt ändern	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Kanal ändern	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Warteschlange ändern	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Warteschlangenmanager ändern	hlq.ALTER.QMGR	ALTER	Keine Prüfung	-
Thema ändern	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Warteschlange löschen	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Authentifizierungsinformationsobjekt erstellen	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Kanal erstellen	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Warteschlange erstellen	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Subskription erstellen	hlq.DEFINE.SUB	ALTER	Keine Prüfung	-
Thema erstellen	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Kanal löschen	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Warteschlange löschen	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER

Tabelle 51. PCF-Befehle, Profile und zugehörige Zugriffsebenen für die IBM MQ Console (Forts.)				
Befehl	**Befehlsprofil für MQCMDS**	**Zugriffsebene für MQCMDS**	**Befehlsressourcenprofil für MQADMIN oder MXADMIN**	**Zugriffsebene für MQADMIN oder MXADMIN**
Subskription löschen	hlq.DELETE.SUB	ALTER	Keine Prüfung	-
Thema löschen	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	Keine Prüfung	-
Objektnamen für Authentifizierungsinformationen anfragen	hlq.DISPLAY.AUTHINFO	READ	Keine Prüfung	-
Kanalinquire	hlq.DISPLAY.CHANNEL	READ	Keine Prüfung	-
Kanalauthentifizierungsdatensätze abgefragt	hlq.DISPLAY.CHLAUTH	READ	Keine Prüfung	-
Kanalinitiator inquire	hlq.DISPLAY.CHINIT	READ	Keine Prüfung	-
Kanalnamen inquire	hlq.DISPLAY.CHANNEL	READ	Keine Prüfung	-
Kanalstatus abgefragt	hlq.DISPLAY.CHSTATUS	READ	Keine Prüfung	-
Warteschlange einfragen	hlq.DISPLAY.QUEUE	READ	Keine Prüfung	-
Warteschlangenmanager abfragen	hlq.DISPLAY.QMGR	READ	Keine Prüfung	-
Warteschlangennamen inquire	hlq.DISPLAY.QUEUE	READ	Keine Prüfung	-
Warteschlangenstatus abgefragt	hlq.DISPLAY.QSTATUS	READ	Keine Prüfung	-
Subskription inquire	hlq.INQUIRE.SUB	READ	Keine Prüfung	-
Subskriptionsstatus abfragen	hlq.INQUIRE.SBSTATUS	READ	Keine Prüfung	-
Thema erfragen	hlq.DISPLAY.TOPIC	READ	Keine Prüfung	-
Themennamen inquire	hlq.DISPLAY.TOPIC	READ	Keine Prüfung	-
Themenstatus inquire	hlq.DISPLAY.TPSTATUS	READ	Keine Prüfung	-
Pingkanal	hlq.PING.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
Cluster aktualisieren	hlq.REFRESH.CLUSTER	ALTER	Keine Prüfung	-
Sicherheit aktualisieren	hlq.REFRESH.SECURITY	ALTER	Keine Prüfung	-
Kanal zurücksetzen	hlq.RESET.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
Kanal auflösen	hlq.RESOLVE.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
Kanalauthentifizierungsdatensatz festlegen	hlq.SET.CHLAUTH	STEUERUNG	Keine Prüfung	-
Kanal starten	hlq.START.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG
Kanal stoppen	hlq.STOP.CHANNEL	STEUERUNG	hlq.CHANNEL.channel	STEUERUNG

z/OS *Profile für die Sicherheit der Befehlsressourcen*

Wenn Sie das Sicherheitschalterprofil für die Befehlsressource nicht definiert haben, müssen Sie Ressourcenprofile für jede Ressource zu der entsprechenden Klasse hinzufügen, da die Sicherheitsprüfung für

Ressourcen, die Befehlen zugeordnet sind, zugeordnet werden soll. Dieselben Sicherheitsprofile steuern sowohl die MQSC-als auch die PCF-Befehle.

Wenn Sie das Sicherheitsschalterprofil für die Befehlsressource nicht definiert haben, `hlq.NO.CMD.RESC.CHECKS`, da Sie die Sicherheitsprüfung für Ressourcen, die mit Befehlen verknüpft sind, aktivieren möchten, müssen Sie:

- Fügen Sie ein Ressourcenprofil in der Klasse **MQADMIN** für jede Ressource hinzu, wenn Sie Profile in Großbuchstaben verwenden.
- Fügen Sie für jede Ressource ein Ressourcenprofil in der Klasse **MXADMIN** hinzu, wenn Sie Profile mit Groß-/Kleinschreibung verwenden.

Dieselben Sicherheitsprofile steuern sowohl die MQSC-als auch die PCF-Befehle.

Profile für die Überprüfung der Befehlsressourcensicherheit haben das folgende Format:

```
hlq.type.resourcename
```

Dabei kann `hlq` entweder `qmgr-name` (Warteschlangenmanagername) oder `qsg-name` (Name der Gruppe mit gemeinsamer Warteschlange) sein.

Ein Profil, dem der Name des Warteschlangenmanagers vorangesetzt ist, steuert den Zugriff auf die Ressourcen, die den Befehlen in diesem Warteschlangenmanager zugeordnet sind. Ein Profil mit dem Namen der Gruppe mit gemeinsamer Warteschlange als Präfix steuert den Zugriff auf die Ressourcen, die den Befehlen auf allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange zugeordnet sind. Dieser Zugriff kann auf einem einzelnen WS-Manager außer Kraft gesetzt werden, indem ein WS-Managerebenenprofil für diese Befehlsressource in diesem Warteschlangenmanager definiert wird.

Wenn Ihr Warteschlangenmanager Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie sowohl die Sicherheit auf der Ebene des Warteschlangenmanagers als auch die Sicherheit auf der Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, sucht IBM MQ zuerst nach einem Profil, das dem Namen des Warteschlangenmanagers vorangestellt ist. Wenn er kein Profil gefunden hat, sucht er nach einem Profil, das dem Namen der Gruppe mit gemeinsamer Warteschlange vorangestellt ist.

Der Name des RACF-Profils zur Überprüfung der Sicherheitsressourcen für die Modellwarteschlange CREDIT.WORTHY im Subsystem CSQ1 lautet:

```
CSQ1.QUEUE.CREDIT.WORTHY
```

Da die Profile für alle Typen von Befehlsressourcen in der Klasse MQADMIN enthalten sind, wird der "Typ" -Teil des Profilnamens im Profil benötigt, um zwischen Ressourcen unterschiedlicher Typen zu unterscheiden, die denselben Namen haben. Der Typ "Typ" des Profilnamens kann CHANNEL, QUEUE, TOPIC, PROCESS oder NAMELIST sein. Ein Benutzer kann z. B. berechtigt sein, hlq.QUEUE.PAYROLL.ONE zu definieren, aber nicht berechtigt, hlq.PROCESS.PAYROLL.ONE zu definieren.

Wenn der Ressourcentyp eine Warteschlange und das Profil auf Ebene der Gruppe mit gemeinsamer Warteschlange ist, wird der Zugriff auf eine oder mehrere lokale Warteschlangen in der Gruppe mit gemeinsamer Warteschlange oder auf eine einzelne gemeinsam genutzte Warteschlange aus einem beliebigen Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange gesteuert.

z/OS MQSC-Befehle, -Profile und die zugehörigen Zugriffsebenen zeigen für jeden IBM MQ MQSC-Befehl die erforderlichen Profile zur Überprüfung der Befehlssicherheit und die zugehörige Zugriffsebene für jeden Profil in der MQCMDS-Klasse.

z/OS PCF-Befehle, -Profile und die zugehörigen Zugriffsebenen zeigen für jeden IBM MQ PCF-Befehl die erforderlichen Profile zur Überprüfung der Befehlssicherheit und die zugehörige Zugriffsebene für jeden Profil in der MQCMDS-Klasse.

z/OS *Sicherheitsprüfung der Befehlsressourcen für Aliaswarteschlangen und ferne Warteschlangen*
Sowohl die Aliaswarteschlange als auch die ferne Warteschlange stellt eine Weiterleitungsfunktion bereit. Hinsichtlich der Sicherheitsprüfung dieser beiden Warteschlangen müssen daher einige zusätzliche Aspekte beachtet werden.

Aliaswarteschlangen

Wenn Sie eine Aliaswarteschlange definieren, werden Sicherheitsprüfungen der Befehlsressourcen ausschließlich für den Namen der Aliaswarteschlange durchgeführt, nicht für die Zielwarteschlange, in die die Aliaswarteschlange aufgelöst wird.

Aliaswarteschlangen können sowohl in lokale als auch in ferne Warteschlangen aufgelöst werden. Wenn Sie Benutzern den Zugriff auf bestimmte lokale oder ferne Warteschlangen verwehren möchten, müssen Sie folgendermaßen vorgehen:

1. Gewähren Sie den Benutzern nicht den Zugriff auf diese lokalen und fernen Warteschlangen.
2. Verhindern Sie, dass Benutzer Aliasnamen für diese Warteschlangen definieren können. Verwehren Sie Benutzern hierzu die Möglichkeit, die Befehle DEFINE QALIAS und ALTER QALIAS auszugeben.

Ferne Warteschlangen

Wenn Sie eine ferne Warteschlange definieren, werden Sicherheitsprüfungen der Befehlsressourcen ausschließlich für den Namen der fernen Warteschlange durchgeführt. Für die Namen der Warteschlangen, die in den Attributen RNAME bzw. XMITQ in der Objektdefinition der fernen Warteschlange angegeben wurden, wird keine Prüfung durchgeführt.

z/OS Sicherheitsprofil RESLEVEL

Sie können in der Klasse MQADMIN oder MXADMIN ein spezielles Profil definieren, um die Anzahl der Benutzer-IDs zu steuern, die auf die API-Ressourcensicherheit überprüft wurden. Dieses Profil wird als RESLEVEL-Profil bezeichnet. Die Auswirkungen dieses Profils auf die Sicherheit der API-Ressource hängt davon ab, wie Sie auf IBM MQ zugreifen.

Wenn eine Anwendung eine Verbindung zu IBM MQ herstellen möchte, überprüft IBM MQ den Zugriff, den die der Verbindung zugeordnete Benutzer-ID auf ein Profil in der Klasse MQADMIN oder MXADMIN mit folgender Bezeichnung hat:

```
hlq.RESLEVEL
```

Dabei kann `hlq` entweder `ssid` (Subsystem-ID) oder `qsg` (ID der Gruppe mit gemeinsamer Warteschlange) sein.

Die Benutzer-IDs, die den einzelnen Verbindungstypen zugeordnet sind, sind:

- Die Benutzer-ID der Verbindungstask für Stapelverbindungen
- Die Benutzer-ID des CICS-Adressraums für CICS-Verbindungen
- Die Benutzer-ID für den Adressraum der IMS-Region für IMS-Verbindungen
- Die Benutzer-ID des Kanalinitiatoradressraums für Kanalinitiatorverbindungen

Achtung: RESLEVEL ist eine sehr leistungsfähige Option; sie kann dazu führen, dass die Umgehung aller Ressourcensicherheitsprüfungen für eine bestimmte Verbindung durchgeführt wird.

Wenn Sie kein RESLEVEL-Profil definiert haben, müssen Sie darauf achten, dass kein anderes Profil in der Klasse MQADMIN hlq.RESLEVEList. Angenommen, Sie haben in MQADMIN ein Profil mit dem Namen hlq. * * und kein Profil hlq.RESLEVEL , Vorsicht vor den Folgen des hlq. * * Profil, da es für die RESLEVEL-Prüfung verwendet wird.

Definieren Sie ein hlq.RESLEVEL-Profil, und setzen Sie die UACC auf NONE, anstatt ein RESLEVEL-Profil überhaupt zu haben. Nehmen Sie möglichst wenige Benutzer oder Gruppen in der Zugriffs-

liste wie möglich vor. Einzelheiten zur Überprüfung des Zugriffs auf RESLEVEL finden Sie unter „Überlegungen zur Protokollierung unter z/OS" auf Seite 277.

Wenn Sie nur die Sicherheit auf Warteschlangenmanagerebene verwenden, führt IBM MQ Prüfungen des Typs RESLEVEL für das Profil `qmgr-name.RESLEVEL` aus. Nur bei der Verwendung der Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange führt IBM MQ die Prüfungen RESLEVEL für das Profil `qsg-name.RESLEVEL` aus. Wenn Sie eine Kombination aus der Sicherheit auf Warteschlangenmanagerebene und auf Ebene der Gruppe mit gemeinsamer Warteschlange verwenden, überprüft IBM MQ zuerst, ob das Profil RESLEVEL auf Warteschlangenmanagerebene vorhanden ist. Wenn kein solches Profil gefunden wird, wird nach einem RESLEVEL-Profil auf Ebene der Gruppe mit gemeinsamer Warteschlange gesucht.

Wenn kein RESLEVEL-Profil gefunden wird, aktiviert IBM MQ die Prüfung der ID für den Job und die Task (oder des alternativen Benutzers) für eine CICS- oder eine IMS-Verbindung. Bei einer Stapelverbindung aktiviert IBM MQ die Prüfung der Benutzer-ID für den Job (oder den alternativen Benutzer). Bei einem Kanalinitiator aktiviert IBM MQ die Prüfung der Benutzer-ID für den Kanal und den Nachrichtenkanalagenten (oder den alternativen Benutzer).

Wenn ein RESLEVEL-Profil vorhanden ist, hängt die Ebene der Prüfung von der Umgebung und der Zugriffsebene für das Profil ab.

Wenn Ihr Warteschlangenmanager ein Mitglied einer Gruppe mit gemeinsamer Warteschlange ist und Sie dieses Profil nicht auf Warteschlangenmanagerebene definieren, müssen Sie beachten, dass möglicherweise ein Profil auf Ebene der Gruppe mit gemeinsamer Warteschlange definiert ist, das sich auf die Stufe der Prüfung auswirkt. Um die Prüfung der beiden Benutzer-IDs zu aktivieren, definieren Sie ein RESLEVEL-Profil (mit den Warteschlangenmanagername oder dem Namen der Gruppe mit gemeinsamer Warteschlange als Präfix) mit dem Wert UACC(NONE) und stellen sicher, dass den relevanten Benutzern kein Zugriff auf dieses Profil erteilt wurde.

Wenn Sie den Zugriff berücksichtigen, den die Benutzer-ID des Kanalinitiators für RESLEVEL hat, müssen Sie sich daran erinnern, dass die vom Kanalinitiator eingerichtete Verbindung auch die von den Kanälen verwendete Verbindung ist. Eine Einstellung, die die Umgehung aller Ressourcensicherheitsprüfungen für die Benutzer-ID des Kanalinitiators bewirkt, umgeht effektiv Sicherheitsprüfungen für alle Kanäle. Wenn der Benutzer-ID-Zugriff des Kanalinitiators auf RESLEVEL etwas anderes als NONE ist, wird nur eine Benutzer-ID (für eine Zugriffsebene von READ oder UPDATE) oder keine Benutzer-IDs (für eine Zugriffsebene von CONTROL oder ALTER) auf Zugriff überprüft. Wenn Sie der Benutzer-ID des Kanalinitiators eine andere Zugriffsebene als NONE für RESLEVEL erteilen, müssen Sie sich vergewissern, dass die Auswirkungen dieser Einstellung auf die Sicherheitsprüfungen für Kanäle verstanden werden.

Die Verwendung des Profils RESLEVEL bedeutet, dass keine normalen Sicherheitsprüfdatensätze verwendet werden. Wenn Sie z. B. UAUDIT für einen Benutzer einlegen, wird der Zugriff auf das Profil hlq.RESLEVEL in MQADMIN nicht protokolliert.

Wenn Sie die RACF-Option WARNING im Profil hlq.RESLEVEL verwenden, werden keine RACF-Warnhinweise für Profile in der Klasse RESLEVEL erstellt.

Die Sicherheitsprüfung für Berichtsnachrichten, wie z. B. CODs, wird durch das RESLEVEL-Profil gesteuert, das der ursprünglichen Anwendung zugeordnet ist. Hat die Benutzer-ID eines Stapeljobs beispielsweise die Berechtigung CONTROL oder ALTER für ein RESLEVEL-Profil, werden alle Ressourcenprüfungen, die vom Stapeljob ausgeführt werden, umgangen, einschließlich der Sicherheitsprüfung von Berichtsnachrichten.

Wenn Sie das Profil RESLEVEL ändern, müssen die Benutzer die Verbindung trennen und die Verbindung erneut herstellen, bevor die Änderung wirksam wird. (Dazu gehört das Stoppen und erneute Starten des Kanalinitiators, wenn der Zugriff auf das Profil RESLEVEL für die Adressraumbenutzer-ID des verteilten Warteschlangenadressbereichs geändert wird.)

Um die RESLEVEL-Prüfung zu inaktivieren, verwenden Sie den Systemparameter RESAUDIT.

z/OS *RESLEVEL und Stapelverbindungen*

Wenn über eine Stapelverbindungen oder Verbindungen dieses Typs auf eine IBM MQ-Ressource zugegriffen wird, muss der Benutzer standardmäßig berechtigt sein, für die entsprechende Operation auf

diese Ressource zuzugreifen. Sie können die Sicherheitsprüfung umgehen, indem Sie eine entsprechende RESLEVEL-Definition definieren.

Gibt an, ob der Benutzer auf der Benutzer-ID basiert, die zum Zeitpunkt der Verbindung verwendet wurde, die gleiche Benutzer-ID, die für die Verbindungsüberprüfung verwendet wird.

Sie können RESLEVEL beispielsweise so konfigurieren, dass bei einem Benutzer, dem Sie vertrauen, über eine Stapelverbindung auf bestimmte Ressourcen zugreift, keine API-Ressourcen-Sicherheitsprüfungen durchgeführt werden. Wenn jedoch ein Benutzer, dem Sie nicht vertrauen, versucht, auf dieselben Ressourcen zuzugreifen, werden die Sicherheitsprüfungen als normal ausgeführt. Sie sollten RESLEVEL aktivieren, um die API-Ressourcensicherheitsüberprüfung nur dann zu umgehen, wenn Sie dem Benutzer und den Programmen, die von diesem Benutzer ausgeführt werden, genügend vertrauen.

In der folgenden Tabelle sind die Prüfungen für Stapelverbindungen aufgeführt.

Tabelle 52. Prüfungen von Stapelverbindungen auf verschiedenen RACF-Zugriffsebenen

RACF-Zugriffsebene	**Stufe der Überprüfung**
KEINE	Ressourcenprüfungen ausgeführt
READ	Ressourcenprüfungen ausgeführt
UPDATE	Ressourcenprüfungen ausgeführt
STEUERUNG	Keine Prüfung.
ALTER	Keine Prüfung.

z/OS *RESLEVEL und Systemfunktionen*

Die Anwendung von RESLEVEL auf die Betriebs- und Steuerkonsolen und auf CSQUTIL.

Die Betriebs- und Steuerkonsolen sowie das Dienstprogramm CSQUTIL sind Batch-Anwendungen, die Anforderungen an den Befehlsserver des Warteschlangenmanagers ausgeben; daher treffen auf sie die Betrachtungen unter „RESLEVEL und Stapelverbindungen” auf Seite 250 zu. Sie können RESLEVEL verwenden, um die Sicherheitsprüfung für die von ihnen verwendeten SYSTEM.COMMAND.INPUT-und SYSTEM.COMMAND.REPLY.MODEL-Warteschlangen zu umgehen, jedoch nicht für die dynamischen Warteschlangen SYSTEM.CSQXCMD.*, SYSTEM.CSQOREXX.* und SYSTEM.CSQUTIL.*.

Da der Befehlsserver integraler Bestandteil des Warteschlangenmanagers ist, werden für ihn keine Verbindungs- oder RESLEVEL-Prüfungen durchgeführt. Um die Sicherheit zu gewährleisten, muss der Befehlsserver daher sicherstellen, dass die Benutzer-ID der anfragenden Anwendung eine Berechtigung zum Öffnen der Warteschlange für Antworten hat. Für die Betriebs- und Steuerkonsolen ist dies SYSTEM.CSQOREXX.*. Für CSQUTIL ist dies SYSTEM.CSQUTIL.*. Benutzer benötigen zusätzlich zu den RESLEVEL-Berechtigungen die Berechtigung für diese Warteschlangen, wie in „Sicherheit der Systemwarteschlange” auf Seite 222 beschrieben.

Andere Anwendungen, die den Befehlsserver verwenden, benennen die jeweilige Empfangswarteschlange für Antworten. Diese Anwendungen können den Befehlsserver täuschen, indem sie mit dem Nachrichtenkontext eine vertrauenswürdigere Benutzer-ID an den Befehlsserver übergeben, so dass dieser Nachrichten in Warteschlangen ohne Berechtigung einreiht. Verwenden Sie das Profil CONTEXT, um dies zu verhindern und den Identitätskontext von Nachrichten in SYSTEM.COMMAND.INPUT zu schützen.

z/OS *RESLEVEL und CICS-Verbindungen*

Bei der Sicherheitsprüfung einer API-Ressource für eine CICS-Verbindung werden standardmäßig zwei Benutzer-IDs überprüft. Sie können ändern, welche Benutzer-IDs überprüft werden, indem Sie ein RESLEVEL-Profil einrichten.

Zunächst wird die Benutzer-ID CICS-Adressraums geprüft. Dabei handelt es sich um die Benutzer-ID auf der Jobkarte des CICS-Jobs oder um die Benutzer-ID, die der von CICS gestarteten Task über die z/OS-Klasse STARTED oder die Tabelle mit gestarteten Prozeduren zugeordnet wurde. (Es handelt sich nicht um CICS DFLTUSER.)

Anschließend wird die Benutzer-ID geprüft, die der CICS-Transaktion zugeordnet ist.

Wenn einer dieser Benutzer-IDs keinen Zugriff auf die Ressource hat, schlägt die Anforderung mit einem Beendigungscode von MQRC_NOT_AUTHORIZED fehl. Sowohl die Benutzer-ID des CICS-Adressraums als auch die Benutzer-ID der Person, die die CICS-Transaktion ausführt, müssen Zugriff auf die Ressourcen au der richtigen Ebene haben.

Auswirkungen von RESLEVEL auf die durchgeführten Prüfungen

Je nachdem, wie Sie Ihr RESLEVEL-Profil konfiguriert haben, können Sie ändern, welche Benutzer-IDs überprüft werden, wenn der Zugriff auf eine Ressource angefordert wird. Weitere Informationen finden Sie unter Tabelle 53 auf Seite 252.

Die geprüften Benutzer-IDs richten sich nach der Benutzer-ID, die bei der Verbindungszeit verwendet wird, also von der Benutzer-ID des CICS-Adressraums. Durch diese Steuerung können Sie die Sicherheitsprüfung von API-Ressourcen für IBM MQ-Anforderungen auf einem System umgehen (z. B. auf Testsystem TESTCICS), diese aber für ein anderes System implementieren (z. B. für Produktionssystem PRODCICS).

Anmerkung: Wenn Sie Ihre CICS-Adressraumbenutzer-ID mit dem Attribut "Vertrauenswürdig" in der Klasse STARTED oder der RACF-Tabelle mit den gestarteten Prozeduren ICHRIN03konfigurieren, überschreibt dies alle Benutzer-IDs, die für den CICS-Adressraum überprüft werden, der durch das Profil RESLEVEL für Ihren Queue Manager festgelegt wurde (d. h. der Queue Manager führt die Sicherheitsprüfungen für den CICS-Adressraum nicht aus). Weitere Informationen finden Sie im Handbuch *CICS Transaction Server for z/OS V3.2 RACF Security Guide*.

In der folgenden Tabelle werden die Prüfungen für CICS-Verbindungen gezeigt.

Tabelle 53. Prüfungen auf verschiedenen RACF-Zugriffsebenen für CICS-Verbindungen

RACF-Zugriffsebene	**Stufe der Überprüfung**
KEINE	IBM MQ überprüft die Benutzer-ID des CICS-Adressraums und die ID des Transaktionsbenutzers.
READ	IBM MQ überprüft nur die Benutzer-ID des CICS-Adressraums.
UPDATE	Wenn die Transaktion in CICS mit RESSEC(YES) definiert ist, überprüft IBM MQ die Benutzer-ID des CICS-Adressraums und die ID des Transaktionsbenutzers.
UPDATE	Wenn die Transaktion in CICS mit RESSEC(NO) definiert ist, überprüft IBM MQ nur die Benutzer-ID des CICS-Adressraums.
CONTROL oder ALTER	IBM MQ überprüft keine Benutzer-IDs.

z/OS *RESLEVEL und IMS-Verbindungen*

Wenn eine API-Ressourcensicherheitsprüfung für eine IMS-Verbindung durchgeführt wird, werden standardmäßig zwei Benutzer-IDs überprüft. Sie können ändern, welche Benutzer-IDs überprüft werden, indem Sie ein RESLEVEL-Profil einrichten.

Wenn eine API-Ressourcensicherheitsprüfung für eine IMS-Verbindung durchgeführt wird, werden standardmäßig zwei Benutzer-IDs überprüft, um festzustellen, ob der Zugriff auf die Ressource zulässig ist.

Die erste Benutzer-ID, die geprüft wird, ist der des Adressraums der IMS-Region. Dies wird entweder aus dem Feld USER von der Jobkarte oder aus der Benutzer-ID, die der Region zugeordnet ist, aus der z/OS-Klasse STARTED oder der Tabelle mit gestarteten Prozeduren (SPT) übernommen.

Die zweite Benutzer-ID, die geprüft wird, ist mit der Arbeit verknüpft, die in der abhängigen Region ausgeführt wird. Sie wird entsprechend dem Typ der abhängigen Region bestimmt (siehe Wie die zweite Benutzer-ID für die IMS(tm)-Verbindung ermittelt wird.

Wenn die erste oder zweite IMS-Benutzer-ID keinen Zugriff auf die Ressource hat, schlägt die Anforderung mit dem Beendigungscode MQRC_NOT_AUTHORIZED fehl.

Durch die Einstellung derIBM MQ-Profile des Typs RESLEVEL kann nicht die Benutzer-ID geändert werden, mit der IMS-Transaktionen aus dem von IBM bereitgestellten MQ-IMS-Auslösemonitorprogramm CSQQTRMN geplant werden. Diese Benutzer-ID ist der PSBNAME des Auslösemonitors, der standardmäßig CSQQTRMN ist.

Auswirkungen von RESLEVEL auf die durchgeführten Prüfungen

Je nachdem, wie Sie Ihr RESLEVEL-Profil konfiguriert haben, können Sie ändern, welche Benutzer-IDs überprüft werden, wenn der Zugriff auf eine Ressource angefordert wird. Folgende Prüfungen sind möglich:

- Überprüfen Sie die Benutzer-ID des Adressraums für die IMS-Region und die zweite Benutzer-ID bzw. die alternative Benutzer-ID.
- Überprüfen Sie nur die Benutzer-ID des Adressraum für die IMS-Region.
- Überprüfen Sie keine Benutzer-IDs.

In der folgenden Tabelle werden die Prüfungen für IMS-Verbindungen gezeigt.

Tabelle 54. Prüfungen auf verschiedenen RACF-Zugriffsebenen für IMS-Verbindungen

RACF-Zugriffsebene	Stufe der Überprüfung
KEINE	Überprüfen Sie die Benutzer-ID des Adressraums für IMS und die zweite Benutzer-ID für IMS oder die alternative Benutzer-ID.
READ	Überprüfen Sie die Benutzer-ID des Adressraums für IMS.
UPDATE	Überprüfen Sie die Benutzer-ID des Adressraums für IMS.
STEUERUNG	Keine Prüfung.
ALTER	Keine Prüfung.

z/OS *RESLEVEL und die Kanalinitiatorverbindung*

Wenn eine API-Ressourcensicherheitsprüfung vom Kanalinitiator durchgeführt wird, werden standardmäßig zwei Benutzer-IDs überprüft. Sie können ändern, welche Benutzer-IDs überprüft werden, indem Sie ein RESLEVEL-Profil einrichten.

Wenn eine API-Ressourcensicherheitsprüfung vom Kanalinitiator vorgenommen wird, werden standardmäßig zwei Benutzer-IDs überprüft, um festzustellen, ob der Zugriff auf die Ressource zulässig ist.

Die geprüften Benutzer-IDs können mit dem Kanalattribut MCAUSER angegeben werden, das vom Netz empfangen wurde, die des Kanalinitiatoradressraums oder die alternative Benutzer-ID für den Nachrichtendeskriptor. Welche Benutzer-IDs überprüft werden, hängt von dem verwendeten Kommunikationsprotokoll und der Einstellung des Kanalattributs PUTAUT ab. Weitere Informationen finden Sie unter „Vom Kanalinitiator verwendete Benutzer-IDs” auf Seite 259.

Wenn einer dieser Benutzer-IDs keinen Zugriff auf die Ressource hat, schlägt die Anforderung mit einem Beendigungscode von MQRC_NOT_AUTHORIZED fehl.

Auswirkungen von RESLEVEL auf die durchgeführten Prüfungen

Je nachdem, wie Sie Ihr RESLEVEL-Profil konfiguriert haben, können Sie ändern, welche Benutzer-IDs überprüft werden, wenn der Zugriff auf eine Ressource angefordert wird, und wie viele geprüft werden.

In der folgenden Tabelle sind die Prüfungen aufgeführt, die für die Verbindung des Kanalinitiators und für alle Kanäle durchgeführt wurden, da sie diese Verbindung verwenden.

Tabelle 55. Prüfungen in verschiedenen RACF-Zugriffsebenen für Verbindungen des Kanalinitiators

RACF-Zugriffsebene	Stufe der Überprüfung
KEINE	Überprüfen Sie zwei Benutzer-IDs.

Tabelle 55. Prüfungen in verschiedenen RACF-Zugriffsebenen für Verbindungen des Kanalinitiators (Forts.)	
RACF-Zugriffsebene	**Stufe der Überprüfung**
READ	Überprüfen Sie eine Benutzer-ID.
UPDATE	Überprüfen Sie eine Benutzer-ID.
STEUERUNG	Keine Prüfung.
ALTER	Keine Prüfung.
Anmerkung: Eine Definition der überprüften Benutzer-IDs finden Sie im Abschnitt „Vom Kanalinitiator verwendete Benutzer-IDs” auf Seite 259	

z/OS *RESLEVEL und gruppeninterne Warteschlangensteuerung*

Wenn eine API-Ressourcensicherheitsprüfung vom gruppeninternen Warteschlangenagenten durchgeführt wird, werden standardmäßig zwei Benutzer-IDs überprüft, um festzustellen, ob der Zugriff auf die Ressource zulässig ist. Sie können ändern, welche Benutzer-IDs überprüft werden, indem Sie ein RESLEVEL-Profil einrichten.

Bei den geprüften Benutzer-IDs kann es sich um die Benutzer-ID handeln, die durch das Attribut IGQUSER des empfangenden Warteschlangenmanagers festgelegt wird, die Benutzer-ID des Warteschlangenmanagers in der Gruppe mit gemeinsamer Warteschlange, die die Nachricht in die Warteschlange SYSTEM.QSG.TRANSMIT.QUEUE eingereiht hat, oder die alternative Benutzer-ID, die im Feld *UserIdentifier* des Nachrichtendeskriptors der Nachricht angegeben ist. Weitere Informationen finden Sie unter „Durch den gruppeninternen Agenten zur Steuerung von Warteschlangen verwendete Benutzer-IDs” auf Seite 263.

Da der gruppeninterne Warteschlangenagent eine interne WS-Manager-Task ist, gibt er keine explizite Verbindungsanforderung aus und wird unter der Benutzer-ID des Warteschlangenmanagers ausgeführt. Der Agent für die gruppeninterne Warteschlangensteuerung wird bei der Initialisierung des Warteschlangenmanagers gestartet. Während der Initialisierung des Agenten für gruppeninterne Warteschlangensteuerung überprüft IBM MQ den Zugriff, den die Benutzer-ID, die dem Warteschlangenmanager zugeordnet ist, in einem Profil in der MQADMIN-Klasse hat:

```
hlq.RESLEVEL
```

Diese Prüfung wird immer ausgeführt, es sei denn, der Switch hlq.NO.SUBSYS.SECURITY wurde festgelegt.

Wenn kein RESLEVEL-Profil vorhanden ist, aktiviert IBM MQ die Prüfung von zwei Benutzer-IDs. Wenn ein RESLEVEL-Profil vorhanden ist, hängt die Überprüfungsstufe von der Zugriffsebene ab, die der Benutzer-ID des Warteschlangenmanagers für das Profil erteilt wurde. Unter Prüfungen auf verschiedenen RACF-Zugriffsebenen für den gruppeninternen Warteschlangenagenten finden Sie Informationen zu den Prüfungen, die für den gruppeninternen Agenten zur Steuerung von Warteschlangen durchgeführt werden.

Tabelle 56. Prüfungen auf verschiedenen RACF-Zugriffsebenen für den gruppeninternen Warteschlangenagenten	
RACF-Zugriffsebene	**Stufe der Überprüfung**
KEINE	Überprüfen Sie zwei Benutzer-IDs.
READ	Überprüfen Sie eine Benutzer-ID.
UPDATE	Überprüfen Sie eine Benutzer-ID.
STEUERUNG	Keine Prüfung.

Tabelle 56. Prüfungen auf verschiedenen RACF-Zugriffsebenen für den gruppeninternen Warteschlangenagenten (Forts.)	
RACF-Zugriffsebene	**Stufe der Überprüfung**
ALTER	Keine Prüfung.
Anmerkung: Eine Definition der überprüften Benutzer-IDs finden Sie im Abschnitt „Durch den gruppeninternen Agenten zur Steuerung von Warteschlangen verwendete Benutzer-IDs” auf Seite 263	

Wenn die Berechtigungen, die dem Profil RESLEVEL für die Benutzer-ID des Warteschlangenmanagers erteilt wurden, geändert werden, muss der gruppeninterne Warteschlangenagent gestoppt und erneut gestartet werden, damit die neuen Berechtigungen wieder aufgenommen werden können. Da es keine Möglichkeit gibt, den gruppeninternen Warteschlangenagenten unabhängig zu stoppen und erneut zu starten, muss der Warteschlangenmanager gestoppt und erneut gestartet werden, um dies zu erreichen.

z/OS *RESLEVEL und die überprüften Benutzer-IDs*

Beispiel für das Festlegen eines RESLEVEL-Profils und Erteilen des Zugriffs auf dieses Profil.

Die Benutzer-ID-Prüfung für den Profilnamen für Stapelverbindungen über Benutzer-IDs, die mit dem Profilnamen für LU 6.2-und TCP/IP-Serververbindungskanäle überprüft wurden zeigen, wie RESLEVEL die Überprüfung der Benutzer-IDs auf verschiedene MQI-Anforderungen beeinflusst.

Beispiel: Sie verfügen über einen Warteschlangenmanager mit dem Namen QM66 mit den folgenden Anforderungen:

- Benutzer WS21B soll von der Ressourcensicherheit ausgeschlossen werden.
- Die gestartete CICS-Task WXNCICS, die mit der Benutzer-ID CICSWXN für den Adressraum ausgeführt wird, führt die vollständige Ressourcenprüfung nur für Transaktionen aus, die mit RESSEC(YES) definiert sind.

Geben Sie zum Definieren des entsprechenden RESLEVEL-Profils den folgenden RACF-Befehl aus:

```
RDEFINE MQADMIN QM66.RESLEVEL UACC(NONE)
```

Geben Sie dann den Benutzern mit den folgenden Befehlen Zugriff auf dieses Profil:

```
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(WS21B) ACCESS(CONTROL)
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(CICSWXN) ACCESS(UPDATE)
```

Wenn Sie diese Änderungen vornehmen, während die Benutzer-IDs mit dem Warteschlangenmanager QM66 verbunden sind, müssen die Benutzer die Verbindung trennen und die Verbindung erneut herstellen, bevor die Änderung stattfindet.

Wenn die Subsystemsicherheit nicht aktiv ist, wenn ein Benutzer eine Verbindung herstellt, während dieser Benutzer noch verbunden ist, die Subsystemsicherheit aktiv wird, wird die vollständige Ressourcensicherheitsüberprüfung auf den Benutzer angewendet. Der Benutzer muss erneut eine Verbindung herstellen, um die korrekte RESLEVEL-Verarbeitung abzurufen.

z/OS Benutzer-IDs für die Sicherheitsprüfung unter z/OS

IBM MQ ruft Sicherheitsprüfungen auf Basis von Benutzer-IDs auf, die Benutzern, Terminals, Anwendungen und anderen Ressourcen zugeordnet sind. Diese Themensammlung enthält eine Liste der Benutzer-IDs, die für die einzelnen Sicherheitschecks verwendet werden.

z/OS *Benutzer-IDs für Verbindungssicherheit*

Die Benutzer-ID, die für die Verbindungssicherheit verwendet wird, hängt vom Typ der Verbindung ab.

Verbindungstyp	Inhalt der Benutzer-ID
Stapelverbindung	Die Benutzer-ID der Verbindungstask. Beispiel: • Die TSO-Benutzer-ID • Die Benutzer-ID, die einem Stapeljob durch den JCL-Parameter USER zugeordnet wurde. • Die Benutzer-ID, die einer gestarteten Task von der Klasse STARTED oder der Tabelle mit den gestarteten Prozeduren zugeordnet wurde.
CICS-Verbindung	Die Benutzer-ID des CICS-Adressraums.
IMS-Verbindung	Die Benutzer-ID des Adressraums für die IMS-Region.
Kanalinitiatorverbindung	Die Benutzer-ID des Kanalinitiatoradressraums.

z/OS *Benutzer-IDs für Befehls-und Befehlsressourcensicherheit*

Die Benutzer-ID, die für die Befehlssicherheit oder die Sicherheit der Befehlsressourcen verwendet wird, hängt von der Position ab, an der der Befehl ausgegeben wird

Ausgestellt von ...	Inhalt der Benutzer-ID
CSQINP1, CSQINP2 oder CSQINPT	Es wird keine Prüfung durchgeführt.
Eingabewarteschlange des Systembefehls	Die Benutzer-ID, die in der *UserIdentifier* des Nachrichtendeskriptors der Nachricht gefunden wurde, die den Befehl enthält. Wenn die Nachricht keine *UserIdentifier* enthält, wird eine Benutzer-ID mit Leerzeichen an den Sicherheitsmanager übergeben.
Konsole	Die Benutzer-ID, die an der Konsole angemeldet ist. Wenn die Konsole nicht angemeldet ist, wird die Standardbenutzer-ID, die vom Systemparameter CMDUSER in CSQ6SYSP festgelegt wurde, festgelegt. Für die Ausgabe von Befehlen aus einer Konsole ist das z/OS-Attribut SYS AUTHORITY erforderlich.
SDSF/TSO-Konsole	TSO-oder Jobbenutzer-ID.
Operations-und Steuerkonsolen	TSO-Benutzer-ID. Wenn Sie die Operationen und Steuerkonsolen verwenden wollen, müssen Sie über die entsprechende Berechtigung zum Absetzen der Befehle verfügen, die den von Ihnen gewünschten Aktionen entsprechen. Darüber hinaus müssen Sie über Lesezugriff auf alle hlq.DISPLAY. verfügen. *object* -Profile in der Klasse 'MQCMDS', da die Anzeigen die verschiedenen DISPLAY-Befehle verwenden, um die Informationen zu erfassen, die sie darstellen.
MGCRE	Wenn MGCRE mit UTOKEN verwendet wird, ist die Benutzer-ID in der UTOKEN. Wenn MGCRE ohne UTOKEN ausgegeben wird, wird die TSO-oder Jobbenutzer-ID verwendet.
CSQ0UTIL	Jobbenutzer-ID.
CSQUTIL	Jobbenutzer-ID.
CSQINPX	Die Benutzer-ID des Adressraums des Kanalinitiators.

z/OS *Benutzer-IDs für Ressourcensicherheit (MQOPEN, MQSUB und MQPUT1)*

In diesem Abschnitt ist der Inhalt der normalen und alternativen Benutzer-IDs für die einzelnen Verbindungstypen aufgeführt. Wie viele Prüfungen durchgeführt werden, wird durch das RESLEVEL-Profil bestimmt. Die überprüfte Benutzer-ID ist die für **MQOPEN**-, **MQSUB**- oder **MQPUT1**-Aufrufe verwendete Benutzer-ID.

Anmerkung: Alle Benutzer-ID-Felder werden genau so überprüft, wie sie empfangen werden. Es finden keine Konvertierungen statt. So sind beispielsweise drei Benutzer-ID-Felder, die "Bob", "BOB" und "bob" enthalten, nicht gleichwertig.

z/OS *Für Stapelverbindungen überprüfte Benutzer-IDs*

Die Benutzer-ID, die für eine Stapelverbindung ausgewählt wurde, hängt davon ab, wie die Task ausgeführt wird und ob eine alternative Benutzer-ID angegeben wurde.

Tabelle 57. Benutzer-ID, die den Profilnamen für Stapelverbindungen überprüft

Alternative Benutzer-ID beim Öffnen angegeben?	Profil 'hlq.ALTERNATE.USER.userid'	Profil 'hlq.CONTEXT.queuename'	Profil 'hlq.resourcename'
Nein	-	TAET	TAET
Ja	TAET	TAET	Alt

Schlüssel:

Alt
Alternative Benutzer-ID.

TAET

- Die Benutzer-ID eines TSO- oder einer z/OS UNIX System Services-Anmeldung.
- Die Benutzer-ID, die einem Stapeljob zugeordnet ist.
- Die Benutzer-ID, die einer gestarteten Task von der Klasse STARTED oder der Tabelle mit den gestarteten Prozeduren zugeordnet wurde.
- Die Benutzer-ID, die der ausgeführten gespeicherten Db2-Prozedur zugeordnet ist.

Ein Stapeljob führt einen MQPUT1 in eine Warteschlange mit dem Namen Q1 aus, wobei RESLEVEL auf READ gesetzt ist und die Überprüfung der alternativen Benutzer-IDs inaktiviert ist.

Unter Prüfungen auf verschiedenen RACF(r)-Zugriffsebenen für Stapelverbindungen und Benutzer-ID, die den Profilnamen für Stapelverbindungen überprüft wird gezeigt, dass die ID des Jobbenutzers mit dem Profil hlq.Q1 abgeglichen wird.

z/OS *Benutzer-IDs, die für CICS-Verbindungen geprüft werden*

Die Benutzer-IDs, die für CICS-Verbindungen geprüft werden, hängen davon an, ob ein oder zwei Prüfungen durchgeführt werden sollen und ob eine alternative Benutzer-ID angegeben ist.

Tabelle 58. Prüfung von Benutzer-IDs für den Profilnamen von Benutzer-IDs vom Typ CICS

Alternative Benutzer-ID beim Öffnen angegeben?	Profil 'hlq.ALTERNATE.USER.userid'	Profil 'hlq.CONTEXT.queuename'	Profil 'hlq.resourcename'
Nein, 1 Prüfung	-	ADS	ADS
Nein, 2 Prüfungen	-	ADS + TXN	ADS + TXN
Ja, 1 Prüfung	ADS	ADS	ADS
Ja, 2 Prüfungen	ADS + TXN	ADS + TXN	ADS + ALT

Schlüssel:

Alt
Alternative Benutzer-ID

ADS
Die Benutzer-ID, die dem CICS-Stapeljob zugeordnet ist, oder, wenn CICS als gestartete Task ausgeführt wird, die über die Klasse STARTED oder die Tabelle mit gestarteten Prozeduren zugeordnet ist.

TXN
Die Benutzer-ID, die der CICS-Transaktion zugeordnet ist. Dies ist normalerweise die Benutzer-ID des Terminalbenutzers, der die Transaktion gestartet hat. Es kann sich dabei um den Benutzer CICS DFLTUSER, ein voreingestelltes Sicherheitsterminal (PRESET) oder um einen manuell angemeldeten Benutzer handeln.

Bestimmen Sie die Benutzer-IDs, die auf die folgenden Bedingungen geprüft werden:

- Die RACF-Zugriffsebene auf das RESLEVEL-Profil für eine Benutzer-ID des CICS-Adressraums ist auf NONE gesetzt.
- Ein MQOPEN -Aufruf wird für eine Warteschlange mit MQOO_OUTPUT und MQOO_PASS_IDENTITY_CONTEXT durchgeführt.

Ermitteln Sie zuerst, wie viele CICS-Benutzer-IDs auf Basis des Zugriffs auf das Profil RESLEVEL durch die Benutzer-ID des CICS-Adressraums geprüft werden. Ab Tabelle 53 auf Seite 252 im Thema „RESLEVEL und CICS-Verbindungen” auf Seite 251 werden zwei Benutzer-IDs überprüft, wenn das RESLEVEL-Profil auf NONE gesetzt ist. Anschließend werden gemäß Tabelle 58 auf Seite 257 die folgenden Prüfungen ausgeführt:

- Das Profil hlq.ALTERNATE.USER.userid wurde nicht überprüft.
- Im Profil hlq.CONTEXT.queuename werden die Benutzer-ID des Adressraums für CICS und die ID des CICS-Transaktionsbenutzers geprüft.
- Im Profil hlq.resourcename werden die Benutzer-ID des Adressraums für CICS und die ID des CICS-Transaktionsbenutzers geprüft.

Dies bedeutet, dass vier Sicherheitsprüfungen für diesen MQOPEN -Aufruf durchgeführt werden.

z/OS *Benutzer-IDs, die für IMS-Verbindungen geprüft werden*

Die für IMS-Verbindungen überprüften Benutzer-IDs hängen davon ab, ob ein oder zwei Prüfungen ausgeführt werden sollen und ob eine alternative Benutzer-ID angegeben ist. Wenn eine zweite Benutzer-ID geprüft wird, hängt dies vom Typ der abhängigen Region ab und von welchen Benutzer-IDs verfügbar sind.

Tabelle 59. Prüfung von Benutzer-IDs für den Profilnamen von Benutzer-IDs vom Typ IMS

Alternative Benutzer-ID beim Öffnen angegeben?	Profil 'hlq.ALTERNATE.USER.userid'	Profil 'hlq.CONTEXT.queuename'	Profil 'hlq.resourcename'
Nein, 1 Prüfung	-	REG	REG
Nein, 2 Prüfungen	-	REG + SEC	REG + SEC
Ja, 1 Prüfung	REG	REG	REG
Ja, 2 Prüfungen	REG + SEC	REG + SEC	REG + ALT

Schlüssel:

Alt
Alternative Benutzer-ID.

REG
Die Benutzer-ID wird normalerweise über die Klasse STARTED oder die Tabelle mit gestarteten Prozeduren festgelegt, kann aber bei der Ausführung von IMS auch über einen übergebenen Job durch den Parameter USER JCL festlegt werden.

SEC
Die zweite Benutzer-ID ist der Arbeit zugeordnet, die in einer abhängigen Region ausgeführt wird. Sie wird in Übereinstimmung mit Tabelle 60 auf Seite 259 festgelegt.

Tabelle 60. Ermittlung der zweiten Benutzer-ID für die IMS-Verbindung

Typen der abhängigen Region	Hierarchie für die Bestimmung der zweiten Benutzer-ID
• BMP-Nachricht gesteuert und erfolgreich GET UNIQUE ausgegeben. • IFP und GET UNIQUE ausgegeben. • MPP.	Benutzer-ID, die der IMS-Transaktion zugeordnet ist, wenn der Benutzer angemeldet ist. LTERM-Name, falls verfügbar. PSBNAME.
• BMP-Nachrichtengesteuerte und erfolgreiche GET UNIQUE nicht ausgegeben. • BMP wird nicht nachrichtengesteuert. • IFP und GET UNIQUE nicht ausgegeben.	Benutzer-ID, die dem Adressraum der abhängigen Region von IMS zugeordnet ist, wenn dieser nicht nur aus Leerzeichen oder Nullen besteht. PSBNAME.

z/OS *Vom Kanalinitiator verwendete Benutzer-IDs*

In dieser Themensammlung werden die Benutzer-IDs beschrieben, die für den Empfang von Kanälen und für Client-MQI-Anforderungen, die über Serververbindungskanäle ausgegeben werden, verwendet und überprüft werden. Informationen werden für TCP/IP und für LU6.2 bereitgestellt.

Sie können den Parameter PUTAUT der empfangenden Kanaldefinition verwenden, um den Typ der verwendeten Sicherheitsprüfung zu bestimmen. Um eine konsistente Sicherheitsprüfung im gesamten IBM MQ-Netz zu erreichen, können sie die Optionen ONLYMCA und ALTMCA verwenden.

Sie können den Befehl DISPLAY CHSTATUS verwenden, um die vom MCA verwendete Benutzer-ID zu ermitteln.

z/OS *Kanäle mit TCP/IP empfangen*

Die geprüften Benutzer-IDs hängen von der PUTAUT-Option des Kanals ab und davon, ob eine oder zwei Prüfungen durchgeführt werden sollen.

Tabelle 61. Benutzer-IDs, die mit dem Profilnamen für TCP/IP-Kanäle geprüft

Option PUTAUT auf Empfänger-oder Requesterkanal angegeben	Profil 'hlq.ALTERNATE.USER.userid'	Profil 'hlq.CONTEXT.queuename'	Profil 'hlq.resourcename'
DEF, 1 Prüfung	-	CHL	CHL
DEF, 2 Prüfungen	-	CHL + MCA	CHL + MCA
CTX, 1 Prüfung	CHL	CHL	CHL
CTX, 2 Prüfungen	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 Prüfung	-	Nachrichtenkanalagent	Nachrichtenkanalagent
ONLYMCA, 2 Prüfungen	-	Nachrichtenkanalagent	Nachrichtenkanalagent
ALTMCA, 1 Prüfung	Nachrichtenkanalagent	Nachrichtenkanalagent	Nachrichtenkanalagent

Tabelle 61. Benutzer-IDs, die mit dem Profilnamen für TCP/IP-Kanäle geprüft (Forts.)			
Option PUTAUT auf Empfänger-oder Requesterkanal angegeben	**Profil 'hlq.ALTERNATE.USER.userid'**	**Profil 'hlq.CONTEXT.queuename'**	**Profil 'hlq.resourcename'**
ALTMCA, 2 Prüfungen	Nachrichtenkanalagent	Nachrichtenkanalagent	MCA + ALT

Schlüssel:

MCA (MCA-Benutzer-ID)
Die Benutzer-ID, die für das Kanalattribut MCAUSER im Empfänger angegeben wurde; wenn dieses Feld leer ist, wird die Benutzer-ID des Kanalinitiatoradressraums der Empfänger-oder Requesterseite verwendet.

CHL (Kanalbenutzer-ID)
Bei TCP/IP wird die Sicherheit vom Übertragungssystem für den Kanal nicht unterstützt. Wenn Transport Layer Security (TLS) verwendet wird und ein digitales Zertifikat vom Partner übergeben wurde, wird die diesem Zertifikat zugeordnete Benutzer-ID (falls installiert) oder die einem passenden, durch den RACF-Zertifikatsnamensfilter (Certificate Name Filtering, CNF) ermittelten Filter zugeordnete Benutzer-ID verwendet. Wenn keine zugeordnete Benutzer-ID gefunden wird oder wenn TLS nicht verwendet wird, wird die Benutzer-ID des Kanalinitiatoradressraums des Empfängers oder des Requesterendes als Kanalbenutzer-ID in Kanälen verwendet, die mit dem Parameter PUTAUT definiert sind, der auf DEF oder CTX gesetzt ist.

Anmerkung: Mithilfe des RACF-Zertifikatsnamensfilters kann dieselbe RACF-Benutzer-ID mehreren fernen Benutzern zugeordnet werden, beispielsweise allen Benutzern in der gleichen Organisationseinheit, die normalerweise über dieselbe Sicherheitsberechtigungen verfügen. Dies bedeutet, dass der Server nicht über eine Kopie des Zertifikats eines jeden möglichen fernen Benutzers auf der ganzen Welt verfügen muss und die Zertifikatsverwaltung und -verteilung erheblich vereinfacht.

Wenn der Parameter PUTAUT auf ONLYMCA oder ALTMCA für den Kanal gesetzt ist, wird die Kanalbenutzer-ID ignoriert und die MCA-Benutzer-ID des Empfängers oder des Requesters verwendet. Dies gilt auch für TCP/IP-Kanäle, die TLS verwenden.

ALT (Alternative Benutzer-ID)
Die Benutzer-ID aus den Kontextinformationen (d. a. das Feld *UserIdentifier*) innerhalb des Nachrichtendeskriptors der Nachricht. Diese Benutzer-ID wird in das Feld *AlternateUserID* im Objektdeskriptor versetzt, bevor ein Aufruf **MQOPEN** oder **MQPUT1** für die Zielzielwarteschlange ausgegeben wird.

z/OS *Kanäle mit LU 6.2 empfangen*
Die geprüften Benutzer-IDs hängen von der PUTAUT-Option des Kanals ab und davon, ob eine oder zwei Prüfungen durchgeführt werden sollen.

Tabelle 62. Benutzer-IDs, die mit dem Profilnamen für LU 6.2-Kanäle geprüft werden			
Option PUTAUT auf Empfänger-oder Requesterkanal angegeben	**Profil 'hlq.ALTERNATE.USER.userid'**	**Profil 'hlq.CONTEXT.queuename'**	**Profil 'hlq.resourcename'**
DEF, 1 Prüfung	-	CHL	CHL
DEF, 2 Prüfungen	-	CHL + MCA	CHL + MCA
CTX, 1 Prüfung	CHL	CHL	CHL
CTX, 2 Prüfungen	CHL + MCA	CHL + MCA	CHL + ALT

Tabelle 62. Benutzer-IDs, die mit dem Profilnamen für LU 6.2-Kanäle geprüft werden (Forts.)			
Option PUTAUT auf Empfänger-oder Requesterkanal angegeben	**Profil 'hlq.ALTERNATE.USER.userid'**	**Profil 'hlq.CONTEXT.queuename'**	**Profil 'hlq.resourcename'**
ONLYMCA, 1 Prüfung	-	Nachrichtenkanalagent	Nachrichtenkanalagent
ONLYMCA, 2 Prüfungen	-	Nachrichtenkanalagent	Nachrichtenkanalagent
ALTMCA, 1 Prüfung	Nachrichtenkanalagent	Nachrichtenkanalagent	Nachrichtenkanalagent
ALTMCA, 2 Prüfungen	Nachrichtenkanalagent	Nachrichtenkanalagent	MCA + ALT

Schlüssel:

MCA (MCA-Benutzer-ID)
Die Benutzer-ID, die für das Kanalattribut MCAUSER im Empfänger angegeben wurde; wenn dieses Feld leer ist, wird die Benutzer-ID des Kanalinitiatoradressraums der Empfänger-oder Requesterseite verwendet.

CHL (Kanalbenutzer-ID)

Anforderer-Server-Kanäle
Wenn der Kanal vom anfordernden Benutzer gestartet wird, gibt es keine Möglichkeit, eine Netzwerkbenutzer-ID (die Kanalbenutzer-ID) zu empfangen.

Wenn der Parameter PUTAUT auf DEF oder CTX im Requesterkanal gesetzt ist, ist die Kanalbenutzer-ID der Adressraum des Kanalinitiators des anfordernden Benutzers, da keine Benutzer-ID vom Netzwerk empfangen wird.

Wenn der Parameter PUTAUT auf ONLYMCA oder ALTMCA gesetzt ist, wird die Kanalbenutzer-ID ignoriert und die MCA-Benutzer-ID des Requesters verwendet.

Andere Kanaltypen
Wenn der Parameter PUTAUT auf dem Empfänger-oder Requesterkanal auf DEF oder CTX gesetzt ist, ist die Kanalbenutzer-ID die Benutzer-ID, die vom Kommunikationssystem empfangen wird, wenn der Kanal initialisiert wird.

- Wenn sich der sendende Kanal auf z/OS befindet, handelt es sich bei der empfangenen Benutzer-ID des Kanals um die Benutzer-ID für den Adressraum des Kanalinitiators des Senders.
- Wenn sich der sendende Kanal auf einer andern Plattform befindet (z. B. AIX), wird die empfangene Kanal-Benutzer-ID normalerweise durch den Parameter USERID der Kanaldefinition bereitgestellt.

Wenn die empfangene Benutzer-ID leer ist oder keine Benutzer-ID empfangen wird, wird eine Kanalbenutzer-ID für Leerzeichen verwendet.

ALT (Alternative Benutzer-ID)
Die Benutzer-ID aus den Kontextinformationen (d. a. das Feld *UserIdentifier*) innerhalb des Nachrichtendeskriptors der Nachricht. Diese Benutzer-ID wird in das Feld *AlternateUserID* im Objektdeskriptor verschoben, bevor ein MQOPEN -oder MQPUT1 -Aufruf für die Zielzielwarteschlange ausgegeben wird.

z/OS *Client-MQI-Anforderungen*
Es können verschiedene Benutzer-IDs verwendet werden, je nachdem, welche Benutzer-IDs und Umgebungsvariablen festgelegt wurden. Diese Benutzer-IDs werden abhängig von der verwendeten PUTAUT-Option und der Angabe, ob eine alternative Benutzer-ID angegeben ist, anhand verschiedener Profile überprüft.

In diesem Abschnitt werden die Benutzer-IDs beschrieben, die für Client-MQI-Anforderungen geprüft werden, die über Serververbindungskanäle für TCP/IP und LU 6.2 ausgegeben wurden. Die MCA-Benutzer-ID und die Kanalbenutzer-ID sind für die in den vorherigen Abschnitten beschriebenen TCP/IP-und LU 6.2-Kanäle.

Für Serververbindungskanäle wird die vom Client empfangene Benutzer-ID verwendet, wenn das Attribut MCAUSER leer ist.

Weitere Informationen finden Sie unter „Zugriffssteuerung für Clients” auf Seite 109.

Verwenden Sie für Client- **MQOPEN**-, **MQSUB**-und **MQPUT1** -Anforderungen die folgenden Regeln, um das Profil zu bestimmen, das überprüft wird:

- Wenn die Anforderung eine alternative Benutzerberechtigung angibt, wird eine Prüfung für die Datei *hlq* .ALTERNATE.USER durchgeführt. *userid* -Profil.
- Wenn die Anforderung die Kontextberechtigung angibt, wird eine Prüfung für die Datei *hlq* .CONTEXT durchgeführt. *queuename* -Profil.
- Für alle **MQOPEN**-, **MQSUB**-und **MQPUT1** -Anforderungen erfolgt eine Prüfung für das Profil *hlq.resourcename* .

Wenn Sie ermittelt haben, welche Profile überprüft werden, verwenden Sie die folgende Tabelle, um festzustellen, welche Benutzer-IDs für diese Profile überprüft werden.

Tabelle 63. Für LU 6.2-und TCP/IP-Server-Verbindungskanäle abgecheckter Benutzer-IDs

Option PUTAUT auf dem Serververbindungskanal angegeben	Alternative Benutzer-ID beim Öffnen angegeben?	Profil 'hlq.ALTERNATE.USER.userid'	Profil 'hlq.CONTEXT.queuename'	Profil 'hlq.resourcename'
DEF, 1 Prüfung	Nein	-	CHL	CHL
DEF, 1 Prüfung	Ja	CHL	CHL	CHL
DEF, 2 Prüfungen	Nein	-	CHL + MCA	CHL + MCA
DEF, 2 Prüfungen	Ja	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 Prüfung	Nein	-	Nachrichtenkanalagent	Nachrichtenkanalagent
ONLYMCA, 1 Prüfung	Ja	Nachrichtenkanalagent	Nachrichtenkanalagent	Nachrichtenkanalagent
ONLYMCA, 2 Prüfungen	Nein	-	Nachrichtenkanalagent	Nachrichtenkanalagent
ONLYMCA, 2 Prüfungen	Ja	Nachrichtenkanalagent	Nachrichtenkanalagent	MCA + ALT

Schlüssel:

MCA (MCA-Benutzer-ID)
Die Benutzer-ID, die für das Kanalattribut MCAUSER in der Server-Verbindung angegeben wurde. Ist dieses Feld leer, wird die Benutzer-ID des Kanalinitiatoradressraums verwendet.

CHL (Kanalbenutzer-ID)
Bei TCP/IP wird die Sicherheit vom Übertragungssystem für den Kanal nicht unterstützt. Wenn Transport Layer Security (TLS) verwendet wird und ein digitales Zertifikat vom Partner übergeben wurde, wird die diesem Zertifikat zugeordnete Benutzer-ID (falls installiert) oder die einem passenden, durch den RACF-Zertifikatsnamensfilter (Certificate Name Filtering, CNF) ermittelten Filter zugeordnete Benutzer-ID verwendet. Wenn keine zugeordnete Benutzer-ID gefunden wird oder wenn TLS nicht verwendet wird, wird die Benutzer-ID des Kanalinitiatoradressraums als Kanalbenutzer-ID in Kanälen verwendet, die mit dem Parameter PUTAUT definiert sind, der auf DEF oder CTX gesetzt ist.

Anmerkung: Mithilfe des RACF-Zertifikatsnamensfilters kann dieselbe RACF-Benutzer-ID mehreren fernen Benutzern zugeordnet werden, beispielsweise allen Benutzern in der gleichen Organisationseinheit, die normalerweise über dieselbe Sicherheitsberechtigungen verfügen. Dies bedeutet, dass der Server nicht über eine Kopie des Zertifikats eines jeden möglichen fernen Benutzers auf der ganzen Welt verfügen muss und die Zertifikatsverwaltung und -verteilung erheblich vereinfacht.

Wenn der Parameter PUTAUT auf ONLYMCA oder ALTMCA für den Kanal gesetzt ist, wird die Kanalbenutzer-ID ignoriert und die MCA-Benutzer-ID des Serververbindungskanals wird verwendet. Dies gilt auch für TCP/IP-Kanäle, die TLS verwenden.

ALT (Alternative Benutzer-ID)
Die Benutzer-ID aus den Kontextinformationen (d. a. das Feld *UserIdentifier*) innerhalb des Nachrichtendeskriptors der Nachricht. Diese Benutzer-ID wird in das Feld *AlternateUserID* im Objekt-oder Subskriptionsdeskriptor verschoben, bevor ein **MQOPEN**-, **MQSUB** -oder **MQPUT1** -Aufruf für die Clientanwendung ausgegeben wird.

z/OS *Beispiel für Kanalinitiator*
Dies ist ein Beispiel dafür, wie Benutzer-IDs für die Überprüfung von RACF-Profilen verwendet werden.

Ein Benutzer führt eine **MQPUT1** -Operation in einer Warteschlange auf dem Warteschlangenmanager QM01 aus, die in eine Warteschlange mit dem Namen QB im Warteschlangenmanager QM02 aufgelöst wird. Die Nachricht wird an einen TCP/IP-Kanal mit dem Namen QM01.TO.QM02 gesendet. RESLEVEL ist auf NONE gesetzt, und die Öffnung wird mit der alternativen Benutzer-ID und der Kontextprüfung ausgeführt. Die Empfängerkanaldefinition hat PUTAUT (CTX), und die MCA-Benutzer-ID ist festgelegt. Welche Benutzer-IDs werden auf dem Empfangskanal verwendet, um die Nachricht in die Warteschlange QB zu stellen?

Antwort: Tabelle 55 auf Seite 253 zeigt, dass zwei Benutzer-IDs geprüft werden, da RESLEVEL auf NONE gesetzt ist.

Tabelle 61 auf Seite 259 zeigt, dass, wenn der Parameter PUTAUT auf CTX gesetzt ist und 2 Prüfungen durchgeführt werden, die Überprüfung der folgenden Benutzer-IDs erfolgt:

- Die Kanalinitiatorbenutzer-ID und die MCAUSER-Benutzer-ID werden mit dem Profil hlq.ALTERNATE.USER.userid verglichen.
- Die Kanalinitiatorbenutzer-ID und die MCAUSER-Benutzer-ID werden mit dem Profil hlq.CONTEXT.queuename verglichen.
- Die Kanalinitiatorbenutzer-ID und die im Nachrichtendeskriptor (MQMD) angegebene alternative Benutzer-ID werden gegen das Profil hlq.Q2 überprüft.

z/OS *Durch den gruppeninternen Agenten zur Steuerung von Warteschlangen verwendete Benutzer-IDs*
Welche Benutzer-IDs beim Öffnen von Zielwarteschlangen durch den gruppeninternen Agenten zur Steuerung von Warteschlangen überprüft werden, wird durch die Werte der WS-Manager-Attribute IGQAUT und IGQUSER bestimmt.

Die folgenden Benutzer-IDs sind möglich:

Benutzer-ID der gruppeninternen Warteschlangensteuerung (IGQ)
Die durch das Attribut IGQUSER definierte Benutzer-ID des empfangenden WS-Managers. Wenn diese aus Leerzeichen besteht, wird die Benutzer-ID des empfangenden WS-Managers verwendet. Da der empfangende WS-Manager jedoch über Zugriffsberechtigung auf alle in ihm definierten Warteschlan-

gen verfügt, werden für die Benutzer-ID des empfangenden WS-Managers keine Sicherheitsprüfungen durchgeführt. In diesem Fall geschieht Folgendes:

- Wenn nur eine Benutzer-ID zu überprüfen ist und es sich hierbei um die Benutzer-ID des empfangenden WS-Managers handelt, werden keine Sicherheitsprüfungen durchgeführt. Dies kann auftreten, wenn IGQAUT auf ONLYIGQ oder ALTIGQ gesetzt ist.
- Wenn zwei Benutzer-IDs zu überprüfen sind und es sich bei einer der beiden um die Benutzer-ID des empfangenden WS-Managers handelt, werden ausschließlich für die andere Benutzer-ID Sicherheitsprüfungen durchgeführt. Dies kann auftreten, wenn IGQAUT auf DEF, CTX oder ALTIGQ gesetzt ist.
- Wenn zwei Benutzer-IDs zu überprüfen sind und es sich bei beiden um Benutzer-IDs des empfangenden WS-Managers handelt, werden keine Sicherheitsprüfungen durchgeführt. Dies kann auftreten, wenn IGQAUT auf ONLYIGQ gesetzt ist.

Benutzer-ID des sendenden WS-Managers (SND)
Die Benutzer-ID des Warteschlangenmanagers in der Gruppe mit gemeinsamer Warteschlange, der die Nachricht in die Warteschlange SYSTEM.QSG.TRANSMIT.QUEUE eingereiht hat.

Alternative Benutzer-ID (ALT)
Die im Feld *UserIdentifier* im Nachrichtendeskriptor der Nachricht angegebene Benutzer-ID.

Tabelle 64. Bei gruppeninterner Warteschlangensteuerung gegen den Profilnamen abgeglichene Benutzer-IDs

Auf dem empfangenden WS-Manager aktivierte Option IGQAUT	Profil 'hlq.ALTERNATE.USER.userid'	Profil 'hlq.CONTEXT.queuename'	Profil 'hlq.resourcename'
DEF, 1 Prüfung	-	SND	SND
DEF, 2 Prüfungen	-	SND +IGQ	SND +IGQ
CTX, 1 Prüfung	SND	SND	SND
CTX, 2 Prüfungen	SND + IGQ	SND +IGQ	SND + ALT
ONLYIGQ, 1 Prüfung	-	IGQ	IGQ
ONLYIGQ, 2 Prüfungen	-	IGQ	IGQ
ALTIGQ, 1 Prüfung	-	IGQ	IGQ
ALTIGQ, 2 Prüfungen	IGQ	IGQ	IGQ + ALT

Schlüssel:

Alt
Alternative Benutzer-ID.

IGQ
IGQ-Benutzer-ID.

SND
Benutzer-ID des sendenden WS-Managers.

z/OS *Leere Benutzer-IDs und UACC-Stufen*

Wenn eine leere Benutzer-ID vorhanden ist, wird ein nicht definierter RACF-Benutzer angemeldet. Erteilen Sie dem nicht definierten Benutzer keinen Zugriff auf die Zugriffsberechtigung.

Leere Benutzer-IDs können vorhanden sein, wenn ein Benutzer Nachrichten mit der Kontextsicherheit oder der alternativen Benutzersicherheit bearbeitet oder wenn eine leere Benutzer-ID an IBM MQ über-

geben wird. Eine leere Benutzer-ID wird z. B. verwendet, wenn eine Nachricht ohne Kontext in die Eingabewarteschlange des Systembefehls geschrieben wird.

Anmerkung: Eine Benutzer-ID von " * " (d. a. ein Stern, gefolgt von sieben Leerzeichen) wird als nicht definierte Benutzer-ID behandelt.

IBM MQ übergibt die leere Benutzer-ID an RACF und es wird ein nicht definierter RACF-Benutzer angemeldet. Alle Sicherheitsprüfungen verwenden dann den Universal Access (UACC) für das relevante Profil. Je nachdem, wie Sie Ihre Zugriffsebenen festgelegt haben, kann die UACC dem nicht definierten Benutzer einen freizugrenzenden Zugriff gewähren.

Wenn Sie beispielsweise diesen RACF-Befehl von TSO ausgeben:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.EVERYONE UACC(UPDATE)
```

definieren Sie ein Profil, mit die benutzerdefinierten z/OS-Benutzer-IDs (die nicht in die Zugriffsliste eingereiht wurden) und die nicht definierte RACF-Benutzer-ID aktiviert werden, um Nachrichten in diese Warteschlange einzureihen und von dort abzurufen.

Um gegen leere Benutzer-IDs zu schützen, müssen Sie Ihre Zugriffsebenen sorgfältig planen und die Anzahl der Personen begrenzen, die die Sicherheit von Kontext und alternativer Benutzer verwenden können. Sie müssen verhindern, dass Personen, die die nicht definierte RACF-Benutzer-ID verwenden, Zugriff auf Ressourcen erhalten, der ihnen nicht zusteht. Sie müssen jedoch gleichzeitig den Zugriff auf Personen mit definierten Benutzer-IDs ermöglichen. Dazu können Sie als Benutzer-ID einen Stern (*) im RACF-Befehl PERMIT angeben und damit allen definierten Benutzer-IDs Zugriff auf Ressourcen erteilen. Daher wird allen nicht definierten Benutzer-IDs (z. B. " * ") der Zugriff verweigert. Mit diesen RACF-Befehlen wird beispielsweise verhindert, dass die nicht definierte RACF-Benutzer-ID Zugriff auf die Warteschlange erhält, um Nachrichten einzureihen oder abzurufen:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY UACC(NONE)
PERMIT Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY CLASS(MQQUEUE) ACCESS(UPDATE) ID(*)
```

z/OS z/OS-Benutzer-ID und Multi-Factor Authentication (MFA)

IBM Multi-Factor Authentication for z/OS ermöglicht z/OS-Sicherheitsadministratoren die Erweiterung der SAF-Authentifizierung, indem identifizierte Benutzer mehrere Authentifizierungsfaktoren (z. B. Kennwort und Verschlüsselungstoken) verwenden müssen, um sich bei einem z/OS-System anzumelden. IBM MFA bietet außerdem Unterstützung für zeitbasierte Technologien zur Kennwortgenerierung wie RSA SecurId.

In IBM MQ ist größtenteils nicht bekannt, wie sich Benutzer bei CICS oder den Batchsystemen anmelden, die IBM MQ steuern. Der Berechtigungsnachweis für die ID des angemeldeten Benutzers ist der z/OS-Task oder dem Adressraum zugeordnet und wird von IBM MQ zur Überprüfung der Berechtigung für Ressourcen verwendet. Für die Benutzer-IDs, die für MFA aktiviert sind, können Berechtigungen für IBM MQ-Ressourcen und die Authentifizierung über PassTickets erteilt werden, die für CICS- und IMS-Bridges verwendet werden.

Wichtig: Besondere Hinweise gelten jedoch bei der Verwendung von Anwendungen, wie z. B. IBM MQ Explorer, in denen die Berechtigungsnachweise für eine Benutzer-ID und ein Kennwort in einem MQCONNX-API-Aufruf mit der Option *MQCSP_AUTH_USER_ID_AND_PWD* übergeben werden. IBM MQ verfügt über keine Funktion, mit der ein zusätzlicher Berechtigungsnachweis in dieser API-Anforderung übergeben werden kann.

Einschränkungen und mögliche Fehlerumgehungen werden in dem folgenden Text beschrieben.

IBM MQ Explorer

Der IBM MQ Explorer kann nicht für die Anmeldung an einem z/OS-System mit einer Benutzer-ID verwendet werden, für die MFA aktiviert ist, da es keine Funktion gibt, mit der ein zweiter Authentifizierungsfaktor vom IBM MQ Explorer an z/OS übergeben werden kann.

Darüber hinaus gibt es zwei verschiedene Verfahren, die vom IBM MQ Explorer verwendet werden, um die Berechtigungsnachweise für eine Benutzer-ID und ein Kennwort erneut zu verwenden, die besondere Aufmerksamkeit erfordern, wenn Einmalkennwörter verwendet werden:

1. IBM MQ Explorer kann Kennwörter in einem verschleierten Format auf dem lokalen System speichern, damit die Anmeldung zu einem späteren Zeitpunkt ausgeführt werden kann. Diese Funktion muss inaktiviert werden, indem der Explorer jedes Mal zur Eingabe des Kennworts aufgefordert wird, wenn eine Verbindung zum z/OS-Warteschlangenmanager hergestellt wird.

   Verwenden Sie dazu die folgende Prozedur:

   a. Wählen Sie **WS-Manager** aus.
   b. Wählen Sie in der angezeigten Liste den Warteschlangenmanager aus, den Sie benötigen, und klicken Sie mit der rechten Maustaste auf diesen Warteschlangenmanager.
   c. Wählen Sie in der angezeigten Menüliste **Verbindungsdetails** aus.
   d. Wählen Sie in der nächsten Menüliste **Eigenschaften** aus und wählen Sie die Registerkarte **Benutzer-ID** aus.

      Stellen Sie sicher, dass Sie das Optionsfeld **Eingabeaufforderung für Kennwort** auswählen.
2. Durch verschieden Operationen im IBM MQ Explorer (z. B. Durchsuchen von Nachrichten in Warteschlangen, Testen von Subskriptionen etc.) wird ein neuer Thread gestartet, der sich mit den Berechtigungsnachweisen bei IBM MQ authentifiziert, die zuerst bei der Anmeldung verwendet wurden. Da der Kennwortberechtigungsnachweis nicht erneut verwendet werden kann, können Sie diese Operationen nicht verwenden.

Es gibt zwei mögliche Workarounds auf der MFA-Konfigurationsebene für diese Probleme:

- Verwenden des Ausschlusses der Anwendungs-ID von MFA, um die IBM MQ-Tasks vollständig aus der MFA-Verarbeitung auszuschließen.

  Geben Sie dazu die folgenden Befehle aus:

  1. ```
     RDEFINE MFADEF MFABYPASS.USERID.chinuser
     ```

     Hierbei steht *chinuser* für die Benutzer-ID des Kanalinitiatoradressraums (die dem Kanalinitiator über die STC-Klasse zugeordnet ist).

  2. ```
     PERMIT MFABYPASS.USERID.chinuser CLASS MFADEF ACCESS(READ) ID(explorer user)
     ```

  Weitere Informationen zu diesem Ansatz finden Sie unter IBM MFA für Anwendungen umgehen.
- Verwenden der externen Unterstützung in MFA, die mit IBM MFA 1.2 eingeführt wurde.. Bei dieser Methode wird der IBM MFA-Web-Server vorauthentifiziert und zusätzlich zu Ihrer Benutzer-ID und Ihrem Kennwort wird die zusätzliche Authentifizierung gemäß der Richtlinie angegeben. Der IBM MFA-Server generiert einen Berechtigungsnachweis für einen Cache-Token, den Sie anschließend im Dialog für die IBM MQ Explorer-Authentifizierung angeben. Der Sicherheitsadministrator kann festlegen, dass dieser Berechtigungsnachweis über einen angemessenen Zeitraum wiederholt wird, damit die normale Verwendung von IBM MQ Explorer ermöglicht wird.

Weitere Informationen zu diesem Ansatz finden Sie unter Einführung in IBM MFA.

z/OS IBM MQ for z/OSSicherheitsmanagement

IBM MQ verwendet eine speicherinterne Tabelle für Informationen, die sich auf jeden Benutzer und die Zugriffsanforderungen von jedem Benutzer beziehen. Um diese Tabelle effizient zu verwalten und die Anzahl der Anforderungen von IBM MQ an den externen Sicherheitsmanager (ESM) zu verringern, sind eine Reihe von Steuerelementen verfügbar.

Diese Steuerelemente sind über die Operationen und Steuerkonsolen und über die IBM MQ-Befehle verfügbar.

z/OS *Überprüfung der Benutzer-ID erneut*

Wenn die RACF-Definition eines Benutzers, der IBM MQ-Ressourcen verwendet, geändert wurde, z. B durch die Verbindung des Benutzers mit einer neuen Gruppe, können Sie den Warteschlangenmanager anweisen, diesen Benutzer beim nächsten Zugriff auf eine IBM MQ-Ressource erneut anzumelden. Sie können dazu den IBM MQ-Befehl RVERIFY SECURITY verwenden.

- Der Benutzer HX0804 erhält und stellt Nachrichten in die PAYROLL-Warteschlangen auf dem Warteschlangenmanager PRD1. HX0804 erfordert jetzt jedoch Zugriff auf einige der PENSION-Warteschlangen auf demselben Warteschlangenmanager (PRD1).
- Der Administrator für die Datensicherheit verbindet Benutzer HX0804 mit der RACF-Gruppe, die den Zugriff auf PENSION-Warteschlangen ermöglicht.
- Damit HX0804 sofort auf die PENSION-Warteschlangen zugreifen kann (also ohne den Warteschlangenmanager PRD1 zu beenden oder auf eine Zeitlimitüberschreitung von HX0804 zu warten), müssen Sie den folgenden IBM MQ-Befehl verwenden:

```
RVERIFY SECURITY(HX0804)
```

Anmerkung: Wenn Sie das Benutzer-ID-Zeitlimit für lange Zeiträume (Tage oder sogar Wochen) inaktivieren, während der Warteschlangenmanager ausgeführt wird, müssen Sie sich daran erinnern, den Befehl RVERIFY SECURITY für alle Benutzer auszuführen, die in dieser Zeit widerrufen oder gelöscht wurden.

z/OS *Benutzer-ID-Zeitlimits*

Sie können festlegen, dass IBM MQ einen Benutzer nach einem bestimmten Inaktivitätszeitraum aus einem Warteschlangenmanager abmeldet.

Wenn ein Benutzer auf eine IBM MQ-Ressource zugreift, versucht der Warteschlangenmanager, diesen Benutzer auf dem Warteschlangenmanager anzumelden (wenn die Sicherheit für das Subsystem aktiv ist). Dies bedeutet, dass der Benutzer beim ESM authentifiziert wird. Diese Benutzer bleibt bei IBM MQ angemeldet, bis der Warteschlangenmanager beendet wird oder das *Zeitlimit* der Benutzer-ID überschritten (die Authentifizierung läuft ab) oder erneut überprüft (erneut authentifiziert) wird.

Wenn ein Benutzer das zulässige Zeitlimit überschritten hat, wird die Benutzer-ID innerhalb des WS-Managers *abgemeldet* und alle sicherheitsrelevanten Informationen, die für diesen Benutzer aufbewahrt werden, werden gelöscht. Das Anmelden und Ausschalten des Benutzers innerhalb des WS-Managers ist für das Anwendungsprogramm oder den Benutzer nicht erkennbar.

Eine Zeitlimitüberschreitung liegt vor, wenn ein Benutzer über einen vordefinierten Zeitraum keine IBM MQ-Ressource verwendet hat. Dieser Zeitraum wird durch den MQSC-Befehl ALTER SECURITY festgelegt.

Im Befehl ALTER SECURITY können zwei Werte angegeben werden:

ZEITLIMIT
: Der Zeitraum in Minuten, in dem eine nicht verwendete Benutzer-ID und die zugehörigen Ressourcen innerhalb des IBM MQ-Warteschlangenmanagers verbleiben können.

INTERVAL
: Der Zeitraum in Minuten zwischen den Prüfungen auf Benutzer-IDs und die zugehörigen Ressourcen, um zu ermitteln, ob der *TIMEOUT* abgelaufen ist.

Wenn der Wert für *TIMEOUT* beispielsweise 30 und der Wert für *INTERVAL* 10 beträgt, überprüft IBM MQ alle 10 Minuten die Benutzer-IDs und die zugehörigen Ressourcen, um festzustellen, ob sie in den letzten 30 Minuten benutzt wurden. Wenn eine Benutzer-ID mit Zeitlimitüberschreitung gefunden wird, wird diese Benutzer-ID innerhalb des Warteschlangenmanagers abgemeldet. Wenn zeitbezogene Ressourceninformationen, die Benutzer-IDs ohne Zeitlimitüberschreitung zugeordnet sind, gefunden werden, werden diese Ressourceninformationen gelöscht. Wenn Sie keine Benutzer-IDs aussetzen möchten, setzen Sie den Wert von *INTERVAL* auf null. Wenn der Wert für *INTERVAL* null ist, wird Speicher, der von Benutzer-IDs und den zugehörigen Ressourcen belegt wird, erst freigegeben, wenn Sie einen Befehl **REFRESH SECURITY** oder **RVERIFY SECURITY** absetzen.

Die Optimierung dieses Werts kann von Bedeutung sein, wenn Sie viele einmalige Benutzer haben. Wenn Sie kleine Intervall-und Zeitlimitwerte festlegen, werden Ressourcen, die nicht mehr benötigt werden, freigegeben.

Anmerkung: Wenn Sie Werte für *INTERVAL* oder *TIMEOUT* verwenden, die nicht die Standardwerte sind, müssen Sie den Befehl bei jedem Start des Warteschlangenmanagers erneut eingeben. Sie können dies automatisch tun, indem Sie den Befehl **ALTER SECURITY** in die Datei CSQINP1 für diesen Warteschlangenmanager einfügen.

z/OS *Sicherheit des Warteschlangenmanagers unter z/OS aktualisieren*

In IBM MQ for z/OS werden RACF-Daten zwischengespeichert, um die Leistung zu verbessern. Wenn Sie bestimmte Sicherheitsklassen ändern, müssen Sie diese zwischengespeicherten Informationen aktualisieren. Aktualisieren Sie die Sicherheit aus Leistungsgründen selten. Sie können auch auswählen, dass nur die TLS-Sicherheitsinformationen aktualisiert werden sollen.

Wenn eine Warteschlange zum ersten Mal (oder zum ersten Mal seit einer Sicherheitsaktualisierung) geöffnet wird, führt IBM MQ eine RACF-Prüfung aus, um die Zugriffsberechtigungen des Benutzers abzurufen und diese Informationen in den Cache zu stellen. Die zwischengespeicherten Daten enthalten Benutzer-IDs und Ressourcen, auf denen die Sicherheitsprüfung ausgeführt wurde. Wenn die Warteschlange vom gleichen Benutzer erneut geöffnet wird, muss IBM MQ keine RACF-Prüfungen ausgeben, wenn die zwischengespeicherten Daten vorhanden sind, was zu einer Leistungsverbesserung beiträgt. Bei einer Aktion zur Sicherheitsaktualisierung werden alle zwischengespeicherten Sicherheitsinformationen gelöscht und IBM MQ wird dazu veranlasst, eine neue Prüfung für RACF vorzunehmen. Bei jedem Hinzufügen, Ändern oder Löschen eines RACF-Ressourcenprofils in einer der Klassen MQADMIN, MXADMIN, MQPROC, MXPROC, MQQUEUE, MXQUEUE, MQNLIST, MXNLIST oder MXTOPIC müssen Sie eine Aktualisierung der Sicherheitsinformationen der Warteschlangenmanager veranlassen, in denen diese Informationen enthalten sind. Geben Sie dazu die folgenden Befehle aus:

- Der RACF-Befehl SETROPTS RACLIST(Klassenname) REFRESH für die Aktualisierung auf RACF-Ebene.
- Der Befehl IBM MQ REFRESH SECURITY, um die Sicherheitsinformationen zu aktualisieren, die vom Warteschlangenmanager gehalten werden. Dieser Befehl muss von jedem Warteschlangenmanager ausgegeben werden, der auf die Profile zugreift, die sich geändert haben. Wenn Sie über eine Gruppe mit gemeinsamer Warteschlange verfügen, können Sie den Befehl mit dem Attribut für den Befehlsbereich an alle Warteschlangenmanager in der Gruppe übertragen.

  Anmerkung: Wenn Sie einen neuen Benutzer mit einer vorhandenen Gruppe verbunden haben, müssen Sie den Befehl IBM MQ RVERIFY SECURITY(userid) ausführen. Mit dem Befehl REFRESH SECURITY (*) kann der Warteschlangenmanager diesen Benutzer nicht erneut signieren, wenn er das nächste Mal versucht, auf eine IBM MQ-Ressource zuzugreifen.

Wenn Sie in einer der IBM MQ-Klassen generische Profile verwenden, müssen Sie beim Ändern, Hinzufügen oder Löschen von generischen Profilen ebenfalls normale RACF-Aktualisierungsbefehle ausgeben. Zum Beispiel SETROPTS GENERIC (Klassenname) REFRESH.

Wenn jedoch ein RACF-Ressourcenprofil hinzugefügt, geändert oder gelöscht wird und die Ressource, auf die es angewendet wird, noch nicht aufgerufen wurde (daher werden keine Informationen im Cache gespeichert), verwendet IBM MQ die neuen RACF-Informationen, ohne dass ein Befehl REFRESH SECURITY ausgegeben wird.

Wenn die RACF-Protokollierung aktiviert ist (z. B. durch Verwendung des RACF-Befehls RALTER AUDIT(access-attempt (audit_access_level))) werden keine Daten zwischengespeichert und deshalb greift IBM MQ bei jeder Prüfung direkt auf den RACF-Datenspeicher zurück. Änderungen werden daher sofort übernommen, und REFRESH SECURITY ist für den Zugriff auf die Änderungen nicht erforderlich. Mithilfe des RACF-Befehls RLIST können Sie überprüfen, ob die RACF-Protokollierung aktiviert ist. Sie könnten z. B. den Befehl

```
RLIST MQQUEUE (qmgr.SYSTEM.COMMAND.INPUT) GEN
```

und erhalten die Ergebnisse

```
CLASS      NAME
-----      ----
MQQUEUE    QP*.SYSTEM.COMMAND.*.** (G)
    AUDITING
    --------
    FAILURES(READ)
```

Dies weist darauf hin, dass die Überwachung festgelegt ist. Weitere Informationen finden Sie in den Handbüchern *z/OS Security Server RACF Auditor's Guide* und *z/OS Security Server RACF Command Language Reference*.

In Abbildung 17 auf Seite 269 sind die Situationen aufgeführt, in denen Sicherheitsinformationen in den Cache gestellt und zwischengespeicherte Informationen verwendet werden.

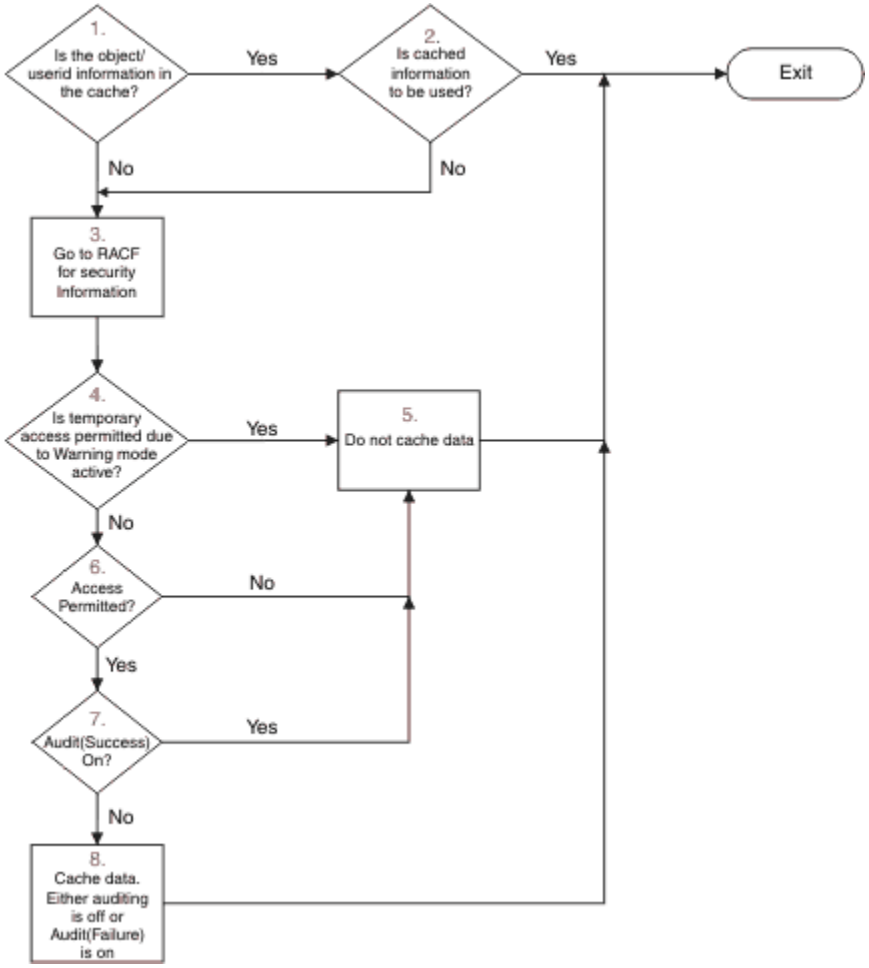


Abbildung 17. Logikablauf für das Zwischenspeichern der IBM MQ-Sicherheit

Wenn Sie Ihre Sicherheitseinstellungen ändern, indem Sie Schalterprofile in den Klassen MQADMIN oder MXADMIN hinzufügen oder löschen, verwenden Sie einen der folgenden Befehle, um diese Änderungen dynamisch zu übernehmen:

REFRESH SECURITY (*)
REFRESH SECURITY (MQADMIN)
REFRESH SECURITY (MXADMIN)

Dies bedeutet, dass Sie neue Sicherheitstypen aktivieren oder inaktivieren können, ohne den WS-Manager erneut starten zu müssen.

Aus Leistungsgründen sind dies die einzigen Klassen, die von dem Befehl REFRESH SECURITY betroffen sind. Sie müssen REFRESH SECURITY nicht verwenden, wenn Sie ein Profil in den Klassen MQCONN oder MQCMDS ändern.

Anmerkung: Eine Aktualisierung der Klasse MQADMIN oder MXADMIN ist nicht erforderlich, wenn Sie ein RESLEVEL-Sicherheitsprofil ändern.

Verwenden Sie die REFRESH SECURITY aus Leistungsgründen so selten wie möglich, idealerweise bei Off-Peak-Zeiten. Sie können die Anzahl der Sicherheitsaktualisierungen verringern, indem Sie Benutzer mit RACF-Gruppen verbinden, die sich bereits in der Zugriffsliste für IBM MQ-Profile befinden. Dies ist einfacher, als einzelne Benutzer in die Zugriffsliste einzufügen. Auf diese Weise ändern Sie den Benutzer und nicht das Ressourcenprofil. Sie können auch RVERIFY SECURITY den entsprechenden Benutzer anstelle der Aktualisierung der Sicherheit verwenden.

Als Beispiel für REFRESH SECURITY wird angenommen, dass Sie die neuen Profile definieren, um den Zugriff auf Warteschlangen zu schützen, die mit INSURANCE.LIFE auf dem Warteschlangenmanager PRMQ beginnen. Sie verwenden die folgenden RACF-Befehle:

```
RDEFINE MQQUEUE PRMQ.INSURANCE.LIFE.** UACC(NONE)
PERMIT PRMQ.INSURANCE.LIFE.** ID(LIFEGRP) ACCESS(UPDATE)
```

Sie müssen den folgenden Befehl ausgeben, damit RACF die gespeicherten Sicherheitsinformationen aktualisiert, z. B.:

```
SETROPTS RACLIST(MQQUEUE) REFRESH
```

Da diese Profile generisch sind, muss RACF zur Aktualisierung der generischen Profile für MQQUEUE veranlasst werden. Beispiel:

```
SETROPTS GENERIC(MQQUEUE) REFRESH
```

Dann müssen Sie mit diesem Befehl WS-Manager PRMQ mitteilen, dass sich die Warteschlangenprofile geändert haben:

```
REFRESH SECURITY(MQQUEUE)
```

SSL/TLS-Sicherheit wird refrecht

Um die zwischengespeicherte Sicht des TLS-Schlüsselrepositorys zu aktualisieren, setzen Sie den Befehl REFRESH SECURITY mit der Option TYPE (SSL) ab. Auf diese Weise können Sie einige TLS-Einstellungen aktualisieren, ohne den Kanalinitiator erneut starten zu müssen.

z/OS *Sicherheitsstatus anzeigen*

Geben Sie den Befehl MQSC DISPLAY SECURITY aus, um den Status der Sicherheitsschalter und andere Sicherheitssteuerungen anzuzeigen.

In der folgenden Abbildung ist die typische Ausgabe des Befehls DISPLAY SECURITY ALL dargestellt.

```
CSQH015I +CSQ1 Security timeout = 54 MINUTES
CSQH016I +CSQ1 Security interval = 12 MINUTES
CSQH030I +CSQ1 Security switches ...
CSQH034I +CSQ1 SUBSYSTEM: ON, 'SQ05.NO.SUBSYS.SECURITY' not found
CSQH032I +CSQ1 QMGR: ON, 'CSQ1.YES.QMGR.CHECKS' found
CSQH031I +CSQ1 QSG: OFF, 'SQ05.NO.QSG.CHECKS' found
CSQH031I +CSQ1 CONNECTION: OFF, 'CSQ1.NO.CONNECT.CHECKS' found
CSQH034I +CSQ1 COMMAND: ON, 'CSQ1.NO.COMMAND.CHECKS' not found
CSQH031I +CSQ1 CONTEXT: OFF, 'CSQ1.NO.CONTEXT.CHECKS' found
CSQH034I +CSQ1 ALTERNATE USER: ON, 'CSQ1.NO.ALTERNATE.USER.CHECKS' not found
CSQH034I +CSQ1 PROCESS: ON, 'CSQ1.NO.PROCESS.CHECKS' not found
CSQH034I +CSQ1 NAMELIST: ON, 'CSQ1.NO.NLIST.CHECKS' not found
CSQH034I +CSQ1 QUEUE: ON, 'CSQ1.NO.QUEUE.CHECKS' not found
CSQH034I +CSQ1 TOPIC: ON, 'CSQ1.NO.TOPIC.CHECKS' not found
CSQH031I +CSQ1 COMMAND RESOURCES: OFF, 'CSQ1.NO.CMD.RESC.CHECKS' found
CSQ9022I +CSQ1 CSQHPDTC ' DISPLAY SECURITY' NORMAL COMPLETION
```

Abbildung 18. Typische Ausgabe des Befehls DISPLAY SECURITY

In diesem Beispiel wird gezeigt, dass im Warteschlangenmanager, der auf den Befehl geantwortet hat, ein Subsystem, ein Befehl, ein alternativer Benutzer, ein Prozess, eine Namensliste und eine Warteschlangensicherheit auf Warteschlangenmanagerebene aktiv ist, aber nicht auf Ebene der Gruppe mit gemeinsamer Warteschlange. Die Verbindung, die Befehlsressource und die Kontextsicherheit sind nicht aktiv. Außerdem wird angezeigt, dass die Benutzer-ID-Zeitlimits aktiv sind, und dass der Warteschlangenmanager alle 12 Minuten nach Benutzer-IDs sucht, die nicht in diesem Warteschlangenmanager für 54 Minuten verwendet wurden, und entfernt sie.

Anmerkung: Mit diesem Befehl wird der aktuelle Sicherheitsstatus angezeigt. Er zeigt nicht unbedingt den aktuellen Status der in RACF definierten Schalterprofile oder den Status der RACFKlassen an. Beispielsweise können die Schalterprofile seit dem letzten Neustart dieses Warteschlangenmanagers oder des Befehls REFRESH SECURITY geändert worden sein.

z/OS Sicherheitsinstallationstasks für z/OS

Nach der Installation und Anpassung von IBM MQ müssen Sie die Prozeduren für gestartete Tasks für RACF berechtigen, den Zugriff auf verschiedene Ressourcen erteilen und RACF-Definitionen konfigurieren. Optional können Sie Ihr System für TLS konfigurieren.

Bei der ersten Installation und Anpassung von IBM MQ müssen Sie die folgenden sicherheitsrelevanten Tasks ausführen:

1. Richten Sie die IBM MQ-Datengruppe und Systemsicherheit folgendermaßen ein:
   - Autorisieren der Prozedur xxxxMSTR für die gestartete Task des Warteschlangenmanagers und der Prozedur xxxxCHIN für die gestartete Task zur verteilten Steuerung von Warteschlangen für die Ausführung unter RACF.
   - Autorisieren des Zugriffs auf WS-Manager-Dateien.
   - Autorisieren des Zugriffs auf Ressourcen für die Benutzer-IDs, die den Warteschlangenmanager und die Dienstprogrammprogramme verwenden.
   - Autorisierender Zugriff für diese Warteschlangenmanager, die die Coupling Facility-Listenstrukturen verwenden.
   - Autorisieren des Zugriffs für die Warteschlangenmanager, die Db2 verwenden sollen.
2. Richten Sie RACF-Definitionen für die IBM MQ-Sicherheit ein.
3. Wenn Sie Transport Layer Security (TLS) verwenden möchten, müssen Sie Ihr System auf die Verwendung von Zertifikaten und Schlüsseln vorbereiten.

z/OS *Sicherheit für IBM MQ for z/OS-Datasets einrichten*

Es gibt viele Typen von IBM MQ-Benutzern. Verwenden Sie RACF, um ihren Zugriff auf Datasets für das System zu steuern.

Die möglichen Benutzer von IBM MQ-Datasets enthalten die folgenden Entitäten:

- Der WS-Manager selbst.
- Kanalinitiator
- IBM MQ-Administratoren, die IBM MQ-Datasets erstellen müssen, führen Dienstprogramme und ähnliche Tasks aus.
- Anwendungsprogrammierer, die die mit IBM MQ bereitgestellten Copybooks verwenden müssen, enthalten Datasets, Makros und ähnliche Ressourcen.
- Anwendungen mit einem oder mehreren der folgenden Schritte:
  - Stapeljobs
  - TSO-Benutzer
  - CICSRegionen
  - IMSRegionen
- Datengruppen CSQOUTX und CSQSNAP
- Dynamische Warteschlangen SYSTEM.CSQXCMD.*

Schützen Sie die IBM MQ-Datasets für alle potenziellen Benutzer mit RACF.

Sie müssen außerdem den Zugriff auf alle Ihre 'CSQINP' -Datensätze steuern.

z/OS *RACF-Autorisierung von gestarteten Taskprozeduren*

Einige IBM MQ-Datasets sind für die exklusive Nutzung durch den Warteschlangenmanager bestimmt. Wenn Sie Ihre IBM MQ-Datasets mithilfe von RACF schützen, müssen Sie ebenfalls der gestarteten Taskprozedur xxxxMSTR für den Warteschlangenmanager und der gestarteten Taskprozedur xxxxCHIN für die verteilte Steuerung von Warteschlangen mithilfe von RACF eine Berechtigung zuweisen. Verwenden Sie dazu die Klasse STARTED. Alternativ können Sie die Tabelle mit gestarteten Prozeduren (ICHRIN03) verwenden, aber dann muss Ihr z/OS-System über IPL gestartet werden, damit die Änderungen wirksam werden.

Weitere Informationen finden Sie im Handbuch *z/OS Security Server RACF System Programmer's Guide*.

Die angegebene RACF-Benutzer-ID muss über die erforderliche Berechtigung auf die Datasets in der gestarteten Taskprozedur verfügen. Wenn Sie beispielsweise eine gestartete Taskprozedur für den Warteschlangenmanager mit der Bezeichnung CSQ1MSTR der RACF-Benutzer-ID QMGRCSQ1 zuordnen, muss die Benutzer-ID QMGRCSQ1 Zugriff auf die z/OS-Ressourcen haben, auf die der Warteschlangenmanager CSQ1 zugreift.

Außerdem muss der Inhalt des Felds GROUP in der Benutzer-ID des Warteschlangenmanagers mit dem Inhalt des Felds GROUP im STARTED-Profil für diesen Warteschlangenmanager identisch sein. Wenn der Inhalt in jedem Gruppenfeld nicht übereinstimmt, wird verhindert, dass die entsprechende Benutzer-ID in das System eingegeben wird. Dadurch wird IBM MQ mit einer nicht definierten Benutzer-ID ausgeführt und infolgedessen aufgrund eines Sicherheitsverstoßes geschlossen.

In den RACF-Benutzer-IDs, die den gestarteten Taskprozeduren für den Warteschlangenmanager und Kanalinitiator zugeordnet sind, darf das Attribut TRUSTED nicht festgelegt sein.

z/OS *Zugriff auf Dateigruppen autorisieren*

Die Dateigruppen für IBM MQ sollten geschützt werden, damit keine unberechtigten Benutzer eine Warteschlangenmanagerinstanz ausführen oder Zugriff auf Warteschlangenmanagerdaten erhalten können. Verwenden Sie dazu den normalen Schutz für z/OS RACF-Datengruppen.

In Tabelle 65 auf Seite 273 wird der RACF-Zugriff zusammengefasst, den die Prozedur für die gestarteten Tasks des Warteschlangenmanagers auf die verschiedenen Datengruppen benötigt.

Tabelle 65. RACF-Zugriff auf Datengruppen, die einem Warteschlangenmanager zugeordnet sind	
RACF-Zugriff	**Datensätze**
READ	• `thlqual.SCSQAUTH` und `thlqual.SCSQANLx` (wobei x für den Buchstaben Ihrer Landessprache steht). • Die Datensätze, auf die von CSQINP1, CSQINP2 und CSQXLIB in der gestarteten Taskprozedur des WS-Managers verwiesen wird. • SMDS-Dateien, deren Eigner andere Warteschlangenmanager in der Gruppe sind. • Protokoll-, BSDS-und Archivprotokolldateien für andere Warteschlangenmanager in der Gruppe.
UPDATE	• Alle Seitengruppen und Protokoll-und BSDS-Dateien. • SMDS-Dateien, deren Eigner ein Warteschlangenmanager ist • SMDS-Datensätze, deren Eigner andere Warteschlangenmanager in der Gruppe sind, für die Strukturen, die der Warteschlangenmanager mit dem Befehl RECOVER CFSTRUCT ausführt.
ALTER	• Alle Archivprotokolldateien.

In Tabelle 66 auf Seite 273 wird der RACF-Zugriff zusammengefasst, den die Prozedur der gestarteten Task für die verteilte Steuerung von Warteschlangen auf den verschiedenen Datengruppen benötigt.

Tabelle 66. RACF-Zugriff auf Datengruppen, die der verteilten Steuerung von Warteschlangen zugeordnet sind	
RACF-Zugriff	**Datensätze**
READ	• thlqual.SCSQAUTH, thlqual.SCSQANLx (wobei x für den Sprachenbuchstabe für Ihre Landessprache steht) und thlqual.SCSQMVR1. • LE-Bibliotheksdatensätze. • Die Dateien, auf die durch CSQXLIB und CSQINPX in der gestarteten Taskprozedur des Kanalinitiators verwiesen wird.
UPDATE	• Datengruppen CSQOUTX und CSQSNAP

Weitere Informationen finden Sie im Handbuch *z/OS Security Server RACF Security Administrator's Guide*.

V 9.2.0 z/OS *Datasets verschlüsseln*

Die IBM MQ-Datasets können mit der z/OS-Dataset-Verschlüsselung zum Schutz der Daten oder aufgrund von gesetzlichen Bestimmungen verschlüsselt werden.

Sie können alle Seitengruppen, aktiven Protokolle, Archivprotokolle und Bootstrap-Dateigruppen (BSDS) mit der z/OS-Dataset-Verschlüsselung schützen.

Achtung: Sie können gemeinsam genutzte Nachrichtendateien (SMDS) nicht mit der z/OS-Dataset-Verschlüsselung von IBM MQ for z/OS 9.1.4 oder früher schützen.

Weitere Informationen finden Sie im Abschnitt zur Vertraulichkeit für ruhende Daten in IBM MQ for z/OS mit der Dataset-Verschlüsselung. weitere Informationen hierzu.

z/OS *IBM MQ for z/OS-Ressourcensicherheit einrichten*

Es gibt viele Typen von IBM MQ-Benutzern. Verwenden Sie RACF zur Steuerung des jeweiligen Zugriffs IBM MQ-Ressourcen.

Die möglichen Benutzer von IBM MQ-Ressourcen wie Warteschlangen und Kanäle enthalten die folgenden Entitäten:

- Der WS-Manager selbst.
- Kanalinitiator
- IBM MQ-Administratoren, die IBM MQ-Datasets erstellen müssen, führen Dienstprogramme und ähnliche Tasks aus.
- Anwendungsprogrammierer, die die mit IBM MQ bereitgestellten Copybooks verwenden müssen, enthalten Datasets, Makros und ähnliche Ressourcen.
- Anwendungen mit einem oder mehreren der folgenden Schritte:
  - Stapeljobs
  - TSO-Benutzer
  - CICSRegionen
  - IMSRegionen
- Datengruppen CSQOUTX und CSQSNAP
- Dynamische Warteschlangen SYSTEM.CSQXCMD.*

Schützen Sie die IBM MQ-Ressourcen für alle potenziellen Benutzer mit RACF. Bedenken Sie insbesondere, dass der Kanalinitiator wie in „Sicherheitsaspekte für den Kanalinitiator unter z/OS” auf Seite 280 beschrieben über Zugriff auf verschiedene Ressourcen verfügen muss, daher muss die Benutzer-ID, unter der er ausgeführt wird, zum Zugriff auf diese Ressourcen berechtigt sein.

Wenn Sie eine Gruppe mit gemeinsamer Warteschlange verwenden, kann der Warteschlangenmanager intern verschiedene Befehle ausgeben, so dass die verwendete Benutzer-ID berechtigt sein muss, solche Befehle auszugeben. Die Befehle sind:

- DEFINE, ALTER und DELETE für jedes Objekt, das QSGDISP (GROUP) hat
- START und STOP CHANNEL für jeden Kanal, der mit CHLDISP verwendet wird (SHARED)

z/OS *z/OS-System für die Verwendung von TLS konfigurieren*

In diesem Abschnitt finden Sie Informationen zum Konfigurieren von IBM MQ for z/OS mit Transport Layer Security (TLS) mithilfe von RACF-Befehlen.

Wenn Sie TLS für die Kanalsicherheit verwenden möchten, müssen Sie eine Reihe von Tasks auf Ihrem System ausführen. (Weitere Informationen zur Verwendung von RACF-Befehlen für Zertifikate und Schlüsselrepositorys (Schlüsselringe) finden Sie unter Mit TLS unter z/OS arbeiten.)

1. Erstellen Sie einen Schlüsselring in RACF, in dem alle Schlüssel und Zertifikate für Ihr System mithilfe des RACF-Befehls RACDCERT gespeichert werden. Beispiel:

   ```
   RACDCERT ID(CHINUSER) ADDRING(QM1RING)
   ```

   Die ID muss entweder die Benutzer-ID des Kanalinitiatoradressraums oder die Benutzer-ID sein, die der Eigner des Schlüsselrings sein soll, wenn es sich um einen gemeinsam genutzten Schlüsselring handeln soll.

2. Erstellen Sie mit dem RACF-Befehl RACDCERT ein digitales Zertifikat für jeden Warteschlangenmanager.

   Die Bezeichnung des Zertifikats muss entweder der Wert des IBM MQ **CERTLABL**-Attributs sein, wenn dieses festgelegt ist, oder der Standardwert `ibmWebSphereMQ`, an den der Name des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels . In diesem Beispiel ist dies `ibmWebSphereMQQM1`.

   Beispiel:

```
RACDCERT ID(USERID) GENCERT
SUBJECTSDN(CN('username') O('IBM') OU('departmentname') C('England'))
WITHLABEL('ibmWebSphereMQQM1')
```

3. Verbinden Sie das Zertifikat in RACF mithilfe des RACF-Befehls RACDCERT mit dem Schlüsselring. Beispiel:

```
RACDCERT CONNECT(ID(USERID) LABEL('ibmWebSphereMQQM1') RING(QM1RING))
CONNECT ID(CHINUSER)
```

   Sie müssen außerdem alle relevanten Unterzeichnerzertifikate (von einer Zertifizierungsstelle) mit dem Schlüsselring verbinden. Dies bedeutet, dass alle Zertifizierungsstellen für das TLS-Zertifikat dieses WS-Managers und alle Zertifizierungsstellen für alle TLS-Zertifikate, mit denen dieser Warteschlangenmanager kommuniziert, ausgeführt werden. Beispiel:

```
RACDCERT ID(CHINUSER)
CONNECT(CERTAUTH LABEL('My CA') RING(QM1RING) USAGE(CERTAUTH))
```

4. Geben Sie auf jedem Ihrer Warteschlangenmanager mit dem IBM MQ-Befehl ALTER QMGR das Schlüsselrepository an, auf das der Warteschlangenmanager verweisen muss. Beispiel: Wenn der Schlüsselring dem Adressraum des Kanalinitiators zugeordnet ist:

```
ALTER QMGR SSLKEYR(QM1RING)
```

   oder wenn Sie einen gemeinsam genutzten Schlüsselring verwenden:

```
ALTER QMGR SSLKEYR(userid/QM1RING)
```

   Hierbei steht *userid* für die Benutzer-ID, die Eigner des gemeinsam genutzten Schlüsselrings ist.

5. Zertifikatswiderrufslisten (Certificate Revocation Lists, CRLs) ermöglichen den Zertifizierungsstellen das Widerrufen von Zertifikaten, die nicht mehr vertrauenswürdig sind. CRLs werden in LDAP-Servern gespeichert. Um auf diese Liste auf dem LDAP-Server zugreifen zu können, müssen Sie zuerst das AUTHINFO-Objekt AUTHTYPE CRLLDAP mithilfe des IBM MQ-Befehls DEFINE AUTHINFO erstellen. Beispiel:

```
DEFINE AUTHINFO(LDAP1)
AUTHTYPE(CRLLDAP)
CONNAME(ldap.server(389))
LDAPUSER('')
LDAPPWD('')
```

   In diesem Beispiel wird die Zertifikatswiderrufsliste in einem öffentlichen Bereich des LDAP-Servers gespeichert, so dass die Felder LDAPUSER und LDAPPWD nicht erforderlich sind.

   Als nächsten müssen Sie Ihr AUTHINFO-Objekt mit dem IBM MQ-Befehl DEFINE NAMELIST in eine Namensliste einfügen. Beispiel:

```
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
```

Ordnen Sie abschließend die Namensliste mit dem IBM MQ-Befehl ALTER QMGR jedem Warteschlangenmanager zu. Beispiel:

```
ALTER QMGR SSLCRLNL(LDAPNL)
```

6. Richten Sie Ihren Warteschlangenmanager mit dem IBM MQ-Befehl ALTER QMGR für die Ausführung von TSL-Aufrufen ein. Dies definiert Server-Subtasks, die nur SSL-Aufrufe verarbeiten, wodurch die normalen Dispatcher die Verarbeitung normal fortsetzen können, ohne dass sie von SSL-Aufrufen betroffen sind. Sie müssen über mindestens zwei dieser Subtasks verfügen. Beispiel:

```
ALTER QMGR SSLTASKS(8)
```

   Diese Änderung tritt nur in Kraft, wenn der Kanalinitiator erneut gestartet wird.

7. Geben Sie mit dem IBM MQ-Befehl DEFINE CHANNEL oder ALTER CHANNEL die Verschlüsselungsspezifikation an, die für jeden Kanal verwendet werden soll. Beispiel:

```
ALTER CHANNEL(LDAPCHL)
CHLTYPE(SDR)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

   Beide Kanalenden müssen dieselbe Chiffrierspezifikation angeben.

z/OS Kanalauthentifizierungsdatensätze in einer QSG verwalten

Kanalauthentifizierungsdatensätze gelten für den Warteschlangenmanager, auf dem sie erstellt werden, sie werden jedoch nicht in der Gruppe mit gemeinsamer Warteschlange (QSG) gemeinsam genutzt. Wenn also alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange dieselben Regeln haben müssen, müssen einige Verwaltungsaufgaben ausgeführt werden, um alle Regeln konsistent zu halten.

1. Fügen Sie die Option CMDSCOPE(*) immer allen SET CHLAUTH -Befehlen hinzu. Dadurch wird der Befehl an alle aktiven Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange gesendet.
2. Verwenden Sie den Befehl DISPLAY CHLAUTH mit der Option CMDSCOPE(*) und analysieren Sie dann die Antworten, um zu prüfen, ob die Datensätze von allen Warteschlangenmanagern identisch sind. Wenn eine Inkonsistenz gefunden wird, kann ein SET CHLAUTH -Befehl ausgegeben werden, der dieselbe Regel enthält, die CMDSCOPE(*) oder CMDSCOPE(*qmgr-name*) enthält.
3. Fügen Sie eine Teildatei zur CSQINP2-Verkettung des Warteschlangenmanagers hinzu (siehe Initialisierungsbefehle), die die vollständigen Regeln enthält. Diese werden als Teil des Initialisierungsprozesses des Warteschlangenmanagers gelesen. Wenn der Befehl SET CHLAUTH die Regel ACTION(ADD) verwendet, wird die Regel nur hinzugefügt, wenn sie nicht vorhanden ist. Die Verwendung von ACTION(REPLACE) ersetzt eine vorhandene Regel, wenn sie bereits vorhanden ist, oder fügt sie hinzu, wenn dies nicht der Fall ist. Das gleiche Mitglied könnte dann in die Verknüpfung CSQINP2 aller Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange platziert werden.
4. Verwenden Sie das Dienstprogramm CSQUTIL (Informationen hierzu finden Sie im Abschnitt Befehle an IBM MQ (COMMAND) absetzen), um die Regeln aus einem Warteschlangenmanager mit der Option MAKEDEF oder MAKEREP zu extrahieren. Anschließend können Sie die Ausgabe mit CSQUTIL in den Zielwarteschlangenmanager wiedergeben.

Zugehörige Konzepte

Kanalauthentifizierungsdatensätze

Die Zugriffsberechtigungen zum Herstellen von Systemverbindungen auf Kanalebene können mithilfe von Kanalauthentifizierungsdatensätzen gezielter gesteuert werden.

z/OS Überlegungen zur Protokollierung unter z/OS

Die normalen RACF-Protokollierungssteuerungen sind für die Durchführung einer Sicherheitsprüfung auf einem Warteschlangenmanager verfügbar. IBM MQ selbst erfasst keine Statistikdaten zur Sicherheit. Die einzigen Statistikdaten sind die Statistiken, die durch die Prüfung erstellt werden können.

RACF-Protokollierung kann auf folgenden Elementen basieren:

- Benutzer-IDs
- Ressourcenklassen
- Profile

Weitere Informationen finden Sie im Handbuch *z/OS Security Server RACF Auditor's Guide*.

Anmerkung: Die Prüfung verschlechtert die Leistung; die mehr Leistung, die Sie implementieren, ist vermindert. Dies ist bei der Verwendung der Option RACF WARNING ebenfalls zu berücksichtigen.

z/OS *RESLEVEL prüfen*

Verwenden Sie den Systemparameter RESAUDIT, um die Erstellung von RESLEVEL-Prüfsätzen zu steuern. Es werden GENERAL-Prüfsätze für RACF erstellt.

Erstellen Sie RESLEVEL-Prüfsätze, indem Sie den Systemparameter RESAUDIT auf YES setzen. Wenn der Parameter RESAUDIT auf NO gesetzt ist, werden keine Prüfsätze erstellt. Weitere Informationen zum Festlegen dieses Parameters enthält der Abschnitt CSQ6SYSP verwenden .

Wenn RESAUDIT auf YES gesetzt ist, werden keine normalen RACF-Prüfsätze verwendet, um bei der RESLEVEL-Prüfung zu ermitteln, welchen Zugriff eine Benutzer-ID für den Adressraum auf das Profil hlq.RESLEVEL hat. Stattdessen fordert IBM MQ an, dass RACF einen GENERAL-Prüfsatz erstellt (Ereignisnummer 27). Diese Prüfungen werden nur zur Verbindungszeit durchgeführt, so dass die Leistungskosten minimal sind.

Achtung: RACFRW ist nicht mehr das empfohlene Dienstprogramm für die Verarbeitung von RACF -Prüfdatensätzen. Sie sollten das RACF SMF data unload verwenden, da dies die bevorzugte Berichtsmethode ist.

Sie können die GENERAL-Prüfsätze von IBM MQ mithilfe des RACF-Berichtsausgabeprogramms (RACFRW) dokumentieren. Sie können die folgenden RACFRW-Befehle verwenden, um den RESLEVEL-Zugriff zu melden:

```
RACFRW
SELECT PROCESS
EVENT GENERAL
LIST
END
```

Ein Beispielbericht aus RACFRW, mit Ausnahme der Felder *Date*, *Time*und *SYSID* , ist in Abbildung 19 auf Seite 278dargestellt.

```
       RACF REPORT - LISTING OF PROCESS RECORDS                                  PAGE   4
                                  E
                                  V  Q
                                  E  U
*JOB/USER *STEP/  --TERMINAL--    N  A
   NAME    GROUP     ID    LVL   T  L

 WS21B    MQMGRP IGJZM000   0    27 0  JOBID=(WS21B 05.111 09:44:57),USERDATA=()
    TRUSTED USER                       AUTH=(NONE),REASON=(NONE)
                                  SESSION=TSOLOGON,TERMINAL=IGJZM000,
                                  LOGSTR='CSQH RESLEVEL CHECK PERFORMED AGAINST PROFI
LE(QM66.RESLEVEL),
                                  CLASS(MQADMIN), ACCESS EQUATES TO (CONTROL)',RESULT=SUC
CESS,MQADMIN
```

Abbildung 19. Beispielausgabe von RACFRW mit RESLEVEL-allgemeinen Prüfsätzen

Wenn Sie die LOGSTR-Daten in dieser Beispielausgabe prüfen, sehen Sie, dass der TSO-Benutzer WS21B über CONTROL-Zugriff auf QM66.RESLEVEL. verfügt. Dies bedeutet, dass alle Ressourcensicherheitsüberprüfungen umgangen werden, wenn der Benutzer WS21B auf QM66-Ressourcen zugreifen kann.

Weitere Informationen zur Verwendung von RACFRW finden Sie im Handbuch *z/OS Security Server RACF Auditor's Guide*.

z/OS Sicherheit anpassen

Wenn Sie die Funktionsweise der IBM MQ-Sicherheit ändern möchten, verwenden Sie den SAF-Exit (ICHRFR00) oder Exits in Ihrem externen Sicherheitsmanager.

Weitere Informationen zu RACF-Exits finden Sie im Handbuch *z/OS Security Server RACROUTE Macro Reference*.

Anmerkung: Da mit IBM MQ die Aufrufe an den ESM optimiert werden, werden RACROUTE-Anforderungen möglicherweise nicht bei beispielsweise jedem Öffnen einer bestimmten Warteschlangen durch einen bestimmten Benutzer vorgenommen.

z/OS Nachrichten zu Sicherheitsverstößen unter z/OS

Ein Sicherheitsverstoß wird durch den Rückkehrcode MQRC_NOT_AUTHORIZED in einem Anwendungsprogramm oder durch eine Nachricht in dem Jobprotokoll angezeigt.

Der Rückkehrcode MQRC_NOT_AUTHORIZED kann aus den folgenden Gründen an ein Anwendungsprogramm zurückgegeben werden:

- Ein Benutzer darf keine Verbindung zum WS-Manager herstellen. In diesem Fall erhalten Sie eine ICH408I-Nachricht im Batch/TSO-, CICS- oder IMS-Jobprotokoll.
- Die Anmeldung eines Benutzers an den Warteschlangenmanager ist fehlgeschlagen, da die Jobbenutzer-ID beispielsweise nicht gültig oder nicht gültig ist, oder die Benutzer-ID oder die alternative Benutzer-ID ist ungültig. Eine oder mehrere dieser Benutzer-IDs sind möglicherweise nicht gültig, da sie widerrufen oder gelöscht wurden. In diesem Fall erhalten Sie eine ICHxxxx-Nachricht und möglicherweise eine IRRxxxx-Nachricht im Jobprotokoll des Warteschlangenmanagers, die die Ursache für den Fehler beim Anmelden gibt. Beispiel:

  ```
  ICH408I USER(NOTDFND ) GROUP(        ) NAME(???                 )
    LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
  IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND
  ```

- Es wurde ein alternativer Benutzer angefordert, aber die Benutzer-ID des Jobs oder der Task hat keinen Zugriff auf die alternative Benutzer-ID. Für diesen Fehler erhalten Sie eine Verstoßnachricht im Jobprotokoll des betreffenden Warteschlangenmanagers.

- Eine Kontextoption wurde verwendet oder impliziert, indem eine Übertragungswarteschlange für die Ausgabe geöffnet wird, aber die Jobbenutzer-ID oder, falls zutreffend, die Task oder die alternative Benutzer-ID keinen Zugriff auf die Kontextoption hat. In diesem Fall wird eine Verstoßnachricht in das Jobprotokoll des betreffenden Warteschlangenmanagers gestellt.
- Ein nicht berechtigter Benutzer hat versucht, auf ein gesichertes WS-Manager-Objekt zuzugreifen, z. B. eine Warteschlange. In diesem Fall wird eine ICH408I-Nachricht für den Verstoß in das Jobprotokoll des betreffenden Warteschlangenmanagers gestellt. Diese Verletzung kann auf den Job oder, falls zutreffend, die Task oder die alternative Benutzer-ID zurückzuführen sein.

Verstöße gegen die Befehlssicherheit und die Sicherheit der Befehlsressourcen können auch im Jobprotokoll des Warteschlangenmanagers gefunden werden.

Wenn in der Nachricht ICH408I (ICH408I) der Jobname des WS-Managers und nicht eine Benutzer-ID angezeigt wird, ist dies normalerweise das Ergebnis einer leeren alternativen Benutzer-ID, die angegeben wird. Beispiel:

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          INSUFFICIENT ACCESS AUTHORITY
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

Sie können herausfinden, wer leere alternative Benutzer-IDs verwenden darf, indem Sie die Zugriffsliste des MQADMIN-Profils hlq.ALTERNATE.USER. -BLANK-überprüfen.

Eine ICH408I-Verstoßnachricht kann auch wie folgt generiert werden:

- Ein Befehl, der ohne Kontext an die Eingabewarteschlange des Systembefehls gesendet wird. Benutzergeschriebene Programme, die in die Eingabewarteschlange des Systembefehls schreiben, sollten immer eine Kontextoption verwenden. Weitere Informationen finden Sie im Abschnitt „Profile für Kontextsicherheit” auf Seite 233.
- Wenn dem Job, der auf die IBM MQ-Ressource zugreift, keine Benutzer-ID zugeordnet ist oder wenn ein IBM MQ-Adapter die Benutzer-ID nicht aus der Adapterumgebung extrahieren kann.

Es können auch Nachrichten zu Verstößen ausgegeben werden, wenn Sie die Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange und auf Warteschlangenmanagerebene verwenden. Sie erhalten möglicherweise Nachrichten, in denen angezeigt wird, dass auf Warteschlangenmanagerebene kein Profil gefunden wurde, aber aufgrund des Profils auf Ebene der Gruppe mit gemeinsamer Warteschlange trotzdem Zugriff gewährt wurde.

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

z/OS Was ist zu tun, wenn der Zugriff nicht ordnungsgemäß zulässig ist oder nicht zulässig ist

Verwenden Sie zusätzlich zu den im Handbuch *z/OS Security Server RACF Security Administrator's Guide* beschriebenen Schritten die folgende Checkliste, wenn die Steuerung des Zugriffs auf eine Ressource nicht ordnungsgemäß zu funktionieren scheint.

- Sind die Schalterprofile korrekt eingestellt?
  - Ist RACF aktiv?
  - Sind die IBM MQ RACF-Klassen installiert und aktiv?

    Verwenden Sie den RACF-Befehl SETROPTS LIST, um dies zu prüfen.

  - Verwenden Sie den IBM MQ-Befehl DISPLAY SECURITY, um den aktuellen Schalterstatus des Warteschlangenmanagers anzuzeigen.
  - Überprüfen Sie die Schalterprofile in der Klasse MQADMIN.

    Verwenden Sie dazu die RACF-Befehle SEARCH und RLIST.
  - Überprüfen Sie die RACF-Schalterprofile erneut, indem Sie den IBM MQ-Befehl REFRESH SECURITY(MQADMIN) ausgeben.
- Hat sich das RACF-Ressourcenprofil geändert? Hat z. B. der allgemeine Zugriff auf das Profil geändert oder hat die Zugriffsliste des Profils geändert?
  - Ist das Profil generisch?

    Ist dies der Fall, geben Sie den RACF-Befehl SETROPTS GENERIC(Klassenname) REFRESH aus.
  - Haben Sie die Sicherheit auf diesem WS-Manager aktualisiert?

    Geben Sie bei Bedarf den RACF-Befehl SETROPTS RACLIST(Klassenname) REFRESH aus.

    Falls erforderlich, setzen Sie den Befehl IBM MQ REFRESH SECURITY(*) ab.
- Hat sich die RACF-Definition des Benutzers geändert? Wurde der Benutzer beispielsweise mit einer neuen Gruppe verbunden oder wurde die Benutzerzugriffsberechtigung entzogen?
  - Haben Sie den Benutzer durch Ausgabe des IBM MQ-Befehls RVERIFY SECURITY(Benutzer-ID) erneut überprüft?
- Werden Sicherheitsüberprüfungen aufgrund von RESLEVEL umgangen?
  - Überprüfen Sie den Zugriff der Verbindungsbenutzer-ID auf das Profil RESLEVEL. Ermitteln Sie die RESLEVEL-Einstellungen mithilfe der RACF-Protokolleinträge.
  - Bei Kanälen müssen Sie daran denken, dass die Zugriffsebene, die die Benutzer-ID des Kanalinitiators auf RESLEVEL (RESLEVEL) hat, von allen Kanälen übernommen wird, so dass eine Zugriffsebene, wie z. B. ALTER, die bewirkt, dass alle Prüfungen umgangen werden, dass Sicherheitsprüfungen für alle Kanäle umgangen werden.
  - Überprüfen Sie bei einer Ausführung über CICS die RESSEC-Einstellung der Transaktion.
  - Wenn RESLEVEL geändert wurde, während ein Benutzer verbunden ist, müssen sie die Verbindung trennen und die Verbindung erneut herstellen, bevor die neue Einstellung RESLEVEL wirksam wird.
- Verwenden Sie Gruppen mit gemeinsamer Warteschlange?
  - Wenn Sie die Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange und auf Warteschlangenmanagerebene verwenden, stellen Sie sicher, dass Sie die korrekten Profile definiert haben. Wenn das WS-Manager-Profil nicht definiert ist, wird eine Nachricht an das Protokoll gesendet, in der mitgeteilt wird, dass das Profil nicht gefunden wurde.
  - Haben Sie eine Kombination von Schaltereinstellungen verwendet, die nicht gültig sind, so dass die vollständige Sicherheitsprüfung eingestellt wurde?
  - Müssen Sie Sicherheitsschalter definieren, um einige der Einstellungen der Gruppe mit gemeinsamer Warteschlange für Ihren Warteschlangenmanager zu überschreiben?
  - Hat ein Profil auf Ebene des Warteschlangenmanagers Vorrang vor einem Profil auf Ebene der Gruppe mit gemeinsamer Warteschlange?

z/OS Sicherheitsaspekte für den Kanalinitiator unter z/OS

Wenn Sie die Ressourcensicherheit in einer Umgebung mit verteilter Steuerung von Warteschlangen verwenden, benötigt der Adressraum des Kanalinitiators den entsprechenden Zugriff auf verschiedene IBM MQ-Ressourcen. Sie können die Integrated Cryptographic Support Facility (ICSF) verwenden, um den Kennwortschutzalgorithmus zu verwenden.

Ressourcensicherheit verwenden

Wenn Sie die Ressourcensicherheit verwenden, sollten Sie die folgenden Punkte in Betracht ziehen, wenn Sie die verteilte Steuerung von Warteschlangen verwenden:

Systemwarteschlangen
Der Adressraum des Kanalinitiators benötigt den RACF-Zugriff UPDATE auf die Systemwarteschlangen, die unter „Sicherheit der Systemwarteschlange" auf Seite 222 aufgeführt sind, und auf alle Zielwarteschlangen des Benutzers sowie auf die Warteschlange für nicht zustellbare Nachrichten (siehe „Sicherheit der Warteschlange für nicht zustellbare" auf Seite 220).

Übertragungswarteschlangen
Der Adressraum des Kanalinitiators benötigt ALTER-Zugriff auf alle Benutzerübertragungswarteschlangen.

Kontextsicherheit
Die Kanalbenutzer-ID (und die Benutzer-ID des Nachrichtenkanalagenten, falls dieser angegeben ist) benötigt den RACF-Zugriff CONTROL auf die hlq.CONTEXT.queuename-Profile in der Klasse MQADMIN. Abhängig vom Profil RESLEVEL benötigt die Kanalbenutzer-ID möglicherweise auch CONTROL-Zugriff auf diese Profile.

Alle Kanäle benötigen CONTROL-Zugriff auf den MQADMIN hlq.CONTEXT. Profil für nicht zustellbare Nachrichten. Alle Kanäle (unabhängig davon, ob sie initiieren oder antworten), können Berichte generieren und benötigen daher CONTROL-Zugriff auf das Profil hlq.CONTEXT.reply-q.

SENDER-, CLUSSDR-und SERVER-Kanäle benötigen CONTROL-Zugriff auf die Profile hlq.CONTEXT.xmit-queue-name, da Nachrichten in die Übertragungswarteschlange gestellt werden können, um den Kanal aufzuwecken, um den Kanal zu beenden.

Anmerkung: Wenn die Kanalbenutzer-ID oder eine RACF-Gruppe, mit der die Kanalbenutzer-ID verbunden ist, über den Zugriff CONTROL oder ALTER auf das Profil hlq.RESLEVEL verfügt, werden für den Kanalinitiator oder einen der zugehörigen Kanäle keine Ressourcenprüfungen ausgeführt.

Weitere Informationen finden Sie unter „Profile für Kontextsicherheit" auf Seite 233 „RESLEVEL und die Kanalinitiatorverbindung" auf Seite 253 und „Benutzer-IDs für die Sicherheitsprüfung unter z/OS" auf Seite 255.

CSQINPX
Wenn Sie die Eingabedatei CSQINPX verwenden, benötigt der Kanalinitiator auch Lesezugriff auf CSQINPX und UPDATE für den Zugriff auf die Datei CSQOUTX und die dynamischen Warteschlangen SYSTEM.CSQXCMD. *.

Verbindungssicherheit
Die Verbindungsanforderungen für den Adressraum des Kanalinitiators verwenden eine Verbindung vom Typ CHIN, für die der entsprechende Zugriffsschutz festgelegt sein muss; siehe „Verbindungssicherheitsprofile für den Kanalinitiator" auf Seite 214.

Datensätze
Für den Adressraum des Kanalinitiators ist der entsprechende Zugriff auf die Datensätze des Warteschlangenmanagers erforderlich; siehe „Zugriff auf Dateigruppen autorisieren" auf Seite 272.

Befehle
Für die Befehle zur verteilten Steuerung von Warteschlangen (z. B. DEFINE CHANNEL, START CHINIT, START LISTENER und andere Kanalbefehle) muss die entsprechende Befehlssicherheit festgelegt sein; siehe Tabelle 49 auf Seite 236.

Wenn Sie eine Gruppe mit gemeinsamer Warteschlange verwenden, kann der Kanalinitiator intern verschiedene Befehle ausgeben, so dass die verwendete Benutzer-ID berechtigt sein muss, solche Befehle auszugeben. Bei diesen Befehlen handelt es sich um START und STOP CHANNEL für jeden Kanal, der mit CHLDISP (SHARED) verwendet wird.

Wenn die PSMODE des WS-Managers nicht DISABLED ist, muss der Kanalinitiator über Lesezugriff auf den Befehl DISPLAY PUBSUB verfügen.

Kanalsicherheit

Für Kanäle, insbesondere Empfänger und Serververbindungen, muss die entsprechende Sicherheit konfiguriert sein; weitere Informationen finden Sie unter „Benutzer-IDs für die Sicherheitsprüfung unter z/OS" auf Seite 255.

Sie können auch das Protokoll Transport Layer Security (TLS) verwenden, um die Sicherheit auf Kanälen zu gewährleisten. Im Abschnitt „TLS-Sicherheitsprotokolle in IBM MQ" auf Seite 26 finden Sie weitere Informationen zur Verwendung von TLS mit IBM MQ.

Im Abschnitt „Zugriffssteuerung für Clients" auf Seite 109 finden Sie außerdem Informationen zur Sicherheit für Serververbindungen.

Benutzer-IDs

Die in „Vom Kanalinitiator verwendete Benutzer-IDs" auf Seite 259 und „Durch den gruppeninternen Agenten zur Steuerung von Warteschlangen verwendete Benutzer-IDs" auf Seite 263 beschriebenen Benutzer-IDs benötigen folgenden Zugriff:

- RACF-Zugriff UPDATE auf die entsprechenden Zielwarteschlangen und die Warteschlange für nicht zustellbare Nachrichten
- RACF-Zugriff CONTROL auf das Profil `hlq.CONTEXT.queuename`, wenn die Kontextprüfung beim Empfänger ausgeführt wird
- Angemessener Zugriff auf die Profile hlq.ALTERNATE.USER.userid, die sie möglicherweise verwenden müssen.
- Für Clients der entsprechende RACF-Zugriff auf die Ressourcen, die verwendet werden sollen.

APPC-Sicherheit

Legen Sie die entsprechende APPC-Sicherheit fest, wenn Sie das Übertragungsprotokoll LU 6.2 verwenden. (Verwenden Sie zum Beispiel die Klasse APPCLU RACF.) Informationen zum Konfigurieren der Sicherheit für APPC finden Sie in den folgenden Handbüchern:

- *z/OS V1R2.0 MVS-Planung: APPC-Management*
- *Multiplatform APPC Configuration Guide*, eine IBM Redbooks-Veröffentlichung

Abgehende Übertragungen verwenden die APPC-Option " SECURITY (SAME) " . Daher werden die Benutzer-ID des Kanalinitiatoradressraums und das zugehörige Standardprofil (RACF GROUP) über das Netz an den Empfänger übergeben. Dabei wird angezeigt, dass die Benutzer-ID bereits geprüft wurde (ALREADYV).

Wenn es sich auf der Empfangsseite ebenfalls um ein z/OS-System handelt, werden die Benutzer-ID und das Profil von APPC geprüft und die Benutzer-ID wird an den Empfängerkanal übergeben und als Kanalbenutzer-ID verwendet.

Wenn der Warteschlangenmanager in einer Umgebung über APPC mit einem anderen Warteschlangenmanager auf dem gleichen oder einem anderen z/OS-System kommuniziert, muss einer der beiden folgenden Punkte zutreffen:

- Die VTAM-Definition für die kommunizierende LU gibt SETACPT (ALREADYV) an.
- Es gibt ein RACF APPCLU-Profil für die Verbindung zwischen logischen Einheiten, das CONVSEC(ALREADYV) angibt.

Sicherheitseinstellungen ändern

Wenn die RACF-Zugriffsebene geändert wird, über die die Kanalbenutzer-ID oder die Benutzer-ID des Nachrichtenkanalagenten auf eine Zielwarteschlange verfügt, werden diese Änderungen nur für neue Objektkennungen (also neue MQOPEN-Aufrufe) für die Zielwarteschlange wirksam. Die Zeiten beim Öffnen und Schließen von MCAs sind variabel. Wenn ein Kanal bereits aktiv ist, wenn eine solche Zugriffsänderung vorgenommen wird, kann der Nachrichtenkanalagenten weiterhin Nachrichten in die Zielwarteschlange stellen, indem der vorhandene Sicherheitszugriff der Benutzer-IDs verwendet wird und nicht der aktualisierte Sicherheitszugriff. Durch das Stoppen und erneute Starten der Kanäle, um die aktualisierte Zugriffsebene zu erzwingen, wird dieses Szenario vermieden.

Automatischer Neustart
Wenn Sie den Kanalinitiator mit dem Automatic Restart Manager (ARM) von z/OS erneut starten, muss die dem XCFAS-Adressraum zugeordnete Benutzer-ID zur Ausgabe des IBM MQ-Befehls START CHINIT berechtigt sein.

Integrated Cryptographic Service Facility (ICSF) verwenden

Der Kanalinitiator kann ICSF verwenden, um eine zufällige Zahl zu generieren, wenn der Kennwortschutzalgorithmus zum Absetzen von Kennwörtern verwendet wird, die über Clientkanäle fließen, wenn TLS nicht verwendet wird. Der Prozess der Generierung einer Zufallszahl wird als *Entropie* bezeichnet.

Wenn die z/OS-Funktion installiert ist, aber ICSF nicht gestartet wurde, wird die Nachricht CSQX213E angezeigt und der Kanalinitiator verwendet STCK für Entropie.

Die Nachricht CSQX213E warnt Sie, dass der Kennwortschutzalgorithmus nicht so sicher ist, wie er sein könnte. Sie können Ihren Prozess jedoch fortsetzen; es gibt keine anderen Auswirkungen auf die Laufzeit.

Wenn die z/OS-Funktion nicht installiert ist, verwendet der Kanalinitiator automatisch STCK.

Anmerkungen:

1. Die Verwendung von ICSF für Entropie generiert mehr zufällige Sequenzen als die Verwendung von STCK.
2. Wenn Sie ICSF starten, müssen Sie den Kanalinitiator erneut starten.
3. ICSF ist für bestimmte CipherSpecs erforderlich. Wenn Sie versuchen, eine dieser CipherSpecs zu verwenden und ICSF nicht installiert ist, empfangen Sie die Nachricht CSQX629E .

z/OS Sicherheit in Clustern für Warteschlangenmanager unter z/OS

Sicherheitsaspekte bei Clustern sind dieselben für Warteschlangenmanager und Kanäle, die nicht in Gruppen zusammengefasst sind. Der Kanalinitiator benötigt Zugriff auf einige zusätzliche Systemwarteschlangen, und einige zusätzliche Befehle benötigen die entsprechende Sicherheitsgruppe.

Sie können die MCA-Benutzer-ID, die Kanalauthentifizierungsdatensätze, TLS und Sicherheitsexits verwenden, um Clusterkanäle zu authentifizieren (wie bei herkömmlichen Kanälen). Die Kanalauthentifizierungsdatensätze oder der Sicherheitsexit, die sich auf den Clusterempfängerkanal beziehen, müssen überprüfen, ob der ferne Warteschlangenmanager den Zugriff auf die Clusterwarteschlangen des Server-WS-Managers zulässt. Sie können die Verwendung der IBM MQ-Clusterunterstützung starten, ohne den vorhandenen Zugriffsschutz für die Warteschlange zu ändern. Sie müssen jedoch anderen Warteschlangenmanagern im Cluster die Schreibzuschreibung in die Warteschlange SYSTEM.CLUSTER.COMMAND.QUEUE erlauben, wenn sie dem Cluster beitreten sollen.

Die IBM MQ-Clusterunterstützung stellt kein Verfahren bereit, mit dem ein Mitglied eines Clusters ausschließlich auf die Rolle als Client beschränkt wird. Aus diesem Grund müssen Sie sicher sein, dass alle Warteschlangenmanager, die Sie in den Cluster zulassen, vertrauenswürden sind. Wenn ein WS-Manager im Cluster eine Warteschlange mit einem bestimmten Namen erstellt, kann er Nachrichten für diese Warteschlange empfangen, unabhängig davon, ob die Anwendung Nachrichten in diese Warteschlange einreihen soll oder nicht.

Wenn Sie die Zugehörigkeit zu einem Cluster einschränken möchten, müssen Sie die gleiche Maßnahme ergreifen, um zu verhindern, dass WS-Manager Verbindungen zu Empfängerkanälen herstellen. Sie schränken die Zugehörigkeit zu einem Cluster ein, indem Sie Kanalauthentifizierungsdatensätze verwenden oder indem Sie ein Sicherheitsexitprogramm auf dem Empfängerkanal schreiben. Sie können auch ein Exitprogramm schreiben, um zu verhindern, dass nicht berechtigte WS-Manager in die Warteschlange SYSTEM.CLUSTER.COMMAND.QUEUE schreiben.

Anmerkung: Es ist nicht ratsam, Anwendungen zuzulassen, um die Warteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE direkt zu öffnen. Es ist auch nicht ratsam, eine Anwendung zuzulassen, um eine beliebige andere Übertragungswarteschlange direkt zu öffnen.

Wenn Sie die Ressourcensicherheit verwenden, sollten Sie zusätzlich zu den unter „Sicherheitsaspekte für den Kanalinitiator unter z/OS” auf Seite 280 beschriebenen Aspekten die folgenden Punkte berücksichtigen:

Systemwarteschlangen
Der Kanalinitiator benötigt den RACF-Zugriff ALTER auf die folgenden Systemwarteschlangen:

- SYSTEM.CLUSTER.COMMAND QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE.

und UPDATE-Zugriff auf SYSTEM.CLUSTER.REPOSITORY.QUEUE

Er benötigt außerdem Lesezugriff auf alle Namenslisten, die für das Clustering verwendet werden.

Befehle
Definieren Sie die entsprechende Befehlssicherheit (wie unter Tabelle 49 auf Seite 236 beschrieben) für die Befehle für die Clusterunterstützung (REFRESH und RESET CLUSTER, SUSPEND und RESUME QMGR).

z/OS Sicherheitsaspekte bei der Verwendung von IBM MQ mit CICS

Alle CICS -Versionen, die von IBM MQ 9.0.0und höher unterstützt werden, verwenden die von CICS bereitgestellte Version des Adapters und der Bridge.

Weitere Informationen zu Sicherheitsaspekten finden Sie unter:

- Sicherheit für den CICS-MQ Adapter.
- Sicherheit für die CICS-MQ -Bridge.

z/OS Sicherheitsaspekte bei der Verwendung von IBM MQ mit IMS

Verwenden Sie die Informationen in diesem Abschnitt, um Ihre Sicherheitsanforderungen bei der Verwendung von IBM MQ mit IMS zu planen.

Verwenden der Klasse OPERCMDS

Wenn Sie Ressourcen mit RACF in der Klasse OPERCMDS schützen, stellen Sie sicher, dass die dem Adressraum Ihres IBM MQ-Warteschlangenmanagers zugeordnete Benutzer-ID über die Berechtigung zur Ausgabe des Befehls MODIFY auf jedem IMS-System verfügt, zu dem eine Verbindung hergestellt werden kann.

Sicherheitsaspekte für die IMS-Brücke

Es gibt vier Aspekte, die Sie bei der Festlegung Ihrer Sicherheitsanforderungen für die IMS-Bridge berücksichtigen müssen:

- Art der Sicherheitsberechtigung, die für die Herstellung einer Verbindung zwischen IBM MQ und IMS erforderlich ist
- Umfang der Sicherheitsprüfung, die in Anwendungen mithilfe der Bridge für den Zugriff auf IMS ausgeführt wird
- Art der IMS-Ressourcen, die für diese Anwendungen zulässig sind
- Welche Berechtigung soll für Nachrichten verwendet werden, die von der Brücke gestellt und erhalten werden

Wenn Sie Ihre Sicherheitsanforderungen für die IMS-Bridge definieren, müssen Sie Folgendes beachten:

- Nachrichten, die über die Brücke übergeben werden, stammen möglicherweise von Anwendungen auf Plattformen, die keine starken Sicherheitsfunktionen bieten.
- Nachrichten, die über die Brücke übergeben werden, stammen möglicherweise aus Anwendungen, die nicht von demselben Unternehmen oder Unternehmen gesteuert werden.

z/OS *Sicherheitsaspekte für die Verbindung mit IMS*

Erteilten Sie der Benutzer-ID des Adressraums für den IBM MQ-Warteschlangenmanager Zugriff auf die OTMA-Gruppe.

Die IMS-Brücke ist ein OTMA-Client. Die Verbindung zu IMS wird mit der Benutzer-ID für den Adressraum des IBM MQ-Warteschlangenmanagers hergestellt. Dies wird normalerweise als Mitglied der gestarteten Taskgruppe definiert. Dieser Benutzer-ID muss der Zugriff auf die OTMA-Gruppe gewährt werden (es sei denn, die Einstellung /SECURE OTMA ist NONE).

Definieren Sie dazu das folgende Profil in der Klasse FACILITY:

```
IMSXCF.xcfgname.mqxcfmname
```

Dabei steht `xcfgname` für den Namen der XCF-Gruppe und `mqxcfmname` für den Namen des XCF-Mitglieds von IBM MQ.

Sie müssen der Benutzer-ID Ihres IBM MQ-Warteschlangenmanagers Lesezugriff auf dieses Profil erteilen.

Anmerkung:

1. Wenn Sie die Berechtigungen in der Klasse FACILITY ändern, müssen Sie den RACF-Befehl SETROPTS RACLIST(FACILITY) REFRESH ausgeben, damit die Änderungen aktiviert werden.
2. Wenn das Profil hlq.NO.SUBSYS.SECURITY in der Klasse MQADMIN vorhanden ist, wird keine Benutzer-ID an IMS übergeben und die Verbindung schlägt fehl, wenn die Einstellung /SECURE OTMA nicht auf NONE gesetzt ist.

z/OS *Anwendungszugriffssteuerung für die IMS-Bridge*

Definieren Sie ein RACF-Profil in der Klasse FACILITY für jedes IMS-System. Erteilen Sie der Benutzer-ID des IBM MQ-Warteschlangenmanagers eine geeignete Zugriffsebene.

Für jedes IMS-System, zu dem die IMS-Bridge eine Verbindung herstellt, können Sie das folgende RACF-Profil in der Klasse „FACILITY“ definieren, um festzustellen, wie viele Sicherheitsprüfungen für jede an das IMS-System übergebene Nachricht ausgeführt werden.

```
IMSXCF.xcfgname.imsxcfmname
```

Dabei ist `xcfgname` der Name der XCF-Gruppe und `imsxcfmname` ist der Name des XCF-Mitglieds für IMS. (Sie müssen ein separates Profil für jedes IMS-System definieren.)

Die Zugriffsebene, die Sie der Benutzer-ID des IBM MQ-Warteschlangenmanagers in diesem Profil erteilen, wird an IBM MQ zurückgegeben, wenn die IMS-Brücke eine Verbindung zu IMS herstellt, und zeigt die in nachfolgenden Transaktionen erforderliche Sicherheitsebene an. Für nachfolgende Transaktionen fordert IBM MQ die entsprechenden Services von RACF an und übergibt dann, wenn die Benutzer-ID berechtigt ist, die Nachricht an IMS.

OTMA unterstützt den IMS-Befehl /SIGN nicht; allerdings können Sie mit IBM MQ die Zugriffsprüfung für jede Nachricht festlegen, um die Implementierung der erforderlichen Steuerungsebene zu ermöglichen.

Die folgenden Zugriffebeneninformationen können zurückgegeben werden:

NONE oder NO PROFILE FOUND
: Diese Werte geben an, dass die maximale Sicherheit erforderlich ist, d. B. die Authentifizierung für jede Transaktion erforderlich ist. Es wird geprüft, ob die im Feld *UserIdentifier* der MQMD-Struktur angegebene Benutzer-ID und das im Feld *Authenticator* der MQIIH-Struktur angegebene Kennwort oder PassTicket in RACF bekannt sind und ob es sich dabei um eine gültige Kombination handelt. Ein UTOKEN wird mit einem Kennwort oder einem PassTicket erstellt und an IMS übergeben. Das UTOKEN wird nicht zwischengespeichert.

Anmerkung: Wenn das Profil hlq.NO.SUBSYS.SECURITY in der Klasse 'MQADMIN' vorhanden ist, überschreibt diese Sicherheitsstufe das, was im Profil definiert ist.

READ
Dieser Wert gibt an, dass die gleiche Authentifizierung unter den folgenden Umständen wie für NONE ausgeführt werden soll:

- Das erste Mal, dass eine bestimmte Benutzer-ID gefunden wird.
- Wenn die Benutzer-ID bereits gefunden wurde, aber das zwischengespeicherte UTOKEN nicht mit einem Kennwort oder PassTicket erstellt wurde

IBM MQ fordert bei Bedarf ein UTOKEN an und übergibt es an IMS.

Anmerkung: Wenn eine Anforderung zum erneuten Prüfen der Sicherheit ausgeführt wurde, gehen alle zwischengespeicherten Informationen verloren, und ein UTOKEN wird angefordert, wenn die Benutzer-ID zum ersten Mal gefunden wird.

UPDATE
Es wird geprüft, ob die im Feld *UserIdentifier* der MQMD-Struktur angegebene Benutzer-ID in RACF bekannt ist.

Es wird ein UTOKEN erstellt und an IMS übergeben. Das UTOKEN wird zwischengespeichert.

CONTROL/ALTER
Diese Werte zeigen an, dass keine UTOKEN für die Sicherheit von Benutzer-IDs in diesem IMS-System bereitgestellt werden müssen. (Sie sollten diese Option wahrscheinlich nur für Entwicklungs-und Testsysteme verwenden.)

Achtung: Beachten Sie, dass die Benutzer-ID, die im Feld *UserIdentifier* der MQMD-Struktur enthalten ist, weiterhin für **CONTROL/ALTER** übergeben wird.

Anmerkung:

1. Dieser Zugriff wird definiert, wenn IBM MQ eine Verbindung zu IMS herstellt, und bleibt für die Dauer der Verbindung bestehen. Um die Sicherheitsstufe zu ändern, muss der Zugriff auf das Sicherheitsprofil geändert und anschließend die Brücke gestoppt und erneut gestartet werden (z. B. OTMA wird gestoppt und erneut gestartet).
2. Wenn Sie die Berechtigungen in der Klasse FACILITY ändern, müssen Sie den RACF-Befehl SETROPTS RACLIST(FACILITY) REFRESH ausgeben, damit die Änderungen aktiviert werden.
3. Sie können ein Kennwort oder ein PassTicket verwenden, aber Sie müssen sich daran erinnern, dass die IMS-Bridge keine Daten verschlüsselt. Weitere Informationen zur Verwendung von PassTickets finden Sie unter „RACF PassTickets im IMS-Header verwenden” auf Seite 287.
4. Einige dieser Ergebnisse können durch die Sicherheitseinstellungen in IMS durch die Verwendung des Befehls /SECURE OTMA beeinflusst werden.
5. Die zwischengespeicherten UTOKEN-Informationen werden solange beibehalten, wie durch die Parameter INTERVAL und TIMEOUT des IBM MQ-Befehls ALTER SECURITY angegeben ist.
6. Die RACF-Option WARNING hat keine Auswirkung auf das Profil IMSXCF.xcfgname.imsxcfmname. Sie wirkt sich nicht auf die erteilte Zugriffsebene aus und es werden keine WARNING-Nachrichten für RACF erstellt.

z/OS *Sicherheitsprüfung unter IMS*

Nachrichten, die über die Brücke übergeben werden, enthalten Sicherheitsinformationen. Die vorgenommenen Sicherheitsprüfungen hängen von der Einstellung des IMS-Befehls /SECURE OTMA ab.

Jede IBM MQ-Nachricht, die über die Bridge übergeben wird, enthält die folgenden Sicherheitsinformationen:

- Eine Benutzer-ID, die im Feld *UserIdentifier* der MQMD-Struktur enthalten ist.
- Der Sicherheitsbereich, der im Feld *SecurityScope* der MQIIH-Struktur enthalten ist (falls die MQIIH-Struktur vorhanden ist)

- Ein UTOKEN (es sei denn, das IBM MQ-Subsystem verfügt über den Zugriff CONTROL oder ALTER auf das entsprechende IMSXCF.xcfgname.imsxcfmname-Profil)

Die vorgenommenen Sicherheitsprüfungen hängen von der Einstellung des IMS-Befehls /SECURE OTMA ab, wie folgt:

/SECURE OTMA NONE
Es werden keine Sicherheitsprüfungen für die Transaktion durchgeführt.

/SECURE OTMA CHECK
Das Feld *UserIdentifier* der MQMD-Struktur wird für eine Überprüfung der Transaktion oder Befehlsberechtigung an IMS übergeben.

Ein ACEE (Accessor Environment Element) wird in die IMS-Steuerregion integriert.

/SECURE OTMA FULL
Das Feld *UserIdentifier* der MQMD-Struktur wird für eine Überprüfung der Transaktion oder Befehlsberechtigung an IMS übergeben.

Ein ACEE wird in die abhängige IMS-Region sowie in die IMS-Steuerregion integriert.

/SECURE OTMA-PROFILE
Das Feld *UserIdentifier* der MQMD-Struktur wird für eine Überprüfung der Transaktion oder Befehlsberechtigung an IMS übergeben.

Mit dem Feld *SecurityScope* in der MQIIH-Struktur wird festgelegt, ob ein ACEE in die abhängige IMS-Region sowie in die Steuerregion integriert wird.

Anmerkung:

1. Wenn Sie die Berechtigungen in der Klasse TIMS oder CIMS oder in den zugeordneten Gruppenklassen GIMS oder DIMS ändern, müssen Sie die folgenden IMS-Befehle ausgeben, um die Änderungen zu aktivieren:
   - /MODIFY PREPARE RACF
   - /MODIFY COMMIT
2. Wenn Sie /SECURE OTMA PROFILE nicht verwenden, wird jeder Wert, der im Feld *SecurityScope* der MQIIH-Struktur angegeben ist, ignoriert.

z/OS *Sicherheitsprüfung durch die IMS-Brücke*

Abhängig von der ausgeführten Aktion werden verschiedene Berechtigungen verwendet.

Wenn die Brücke eine Nachricht einreiht oder erhält, werden die folgenden Berechtigungen verwendet:

Abrufen einer Nachricht aus der Bridge-Warteschlange
Es werden keine Sicherheitsprüfungen ausgeführt.

Ausnahme-oder COA-Berichtsnachricht einschalten
Verwendet die Berechtigung der Benutzer-ID im Feld *UserIdentifier* der MQMD-Struktur.

Angeben einer Antwortnachricht
Verwendet die Berechtigung der Benutzer-ID im Feld *UserIdentifier* der MQMD-Struktur der ursprünglichen Nachricht.

Nachricht in die Warteschlange für nicht zustellbare Nachrichten einschalten
Es werden keine Sicherheitsprüfungen ausgeführt.

Anmerkung:

1. Wenn Sie die Profile der IBM MQ-Klasse ändern, müssen Sie den IBM MQ-Befehl REFRESH SECURITY(*) ausgeben, um die Änderungen zu aktivieren.
2. Wenn Sie die Berechtigung eines Benutzers ändern, müssen Sie den Befehl MQSC RVERIFY SECURITY ausgeben, um die Änderung zu aktivieren.

z/OS *RACF PassTickets im IMS-Header verwenden*

Sie können ein PassTicket anstelle eines Kennworts im IMS-Header verwenden.

Wenn Sie ein PassTicket anstelle eines Kennworts im IMS-Header (MQIIH) verwenden möchten, geben Sie den Anwendungsnamen an, mit dem das PassTicket im Attribut PASSTKTA der Definition STGCLASS für die IMS-Bridge-Warteschlange, an die die Nachricht weitergeleitet werden soll, geprüft wird.

Wenn der Wert für PASSTKTA leer gelassen wird, müssen Sie ein PassTicket generieren lassen. Der Anwendungsname muss in diesem Fall das Format MVSxxxx haben, wobei xxxx für die SMFID des z/OS-Systems steht, auf dem der Zielwarteschlangenmanager ausgeführt wird.

Ein PassTicket wird aus einer Benutzer-ID, dem Zielanwendungsnamen und einem geheimen Schlüssel erstellt. Es handelt sich um einen 8-Byte-Wert, der alphabetische und numerische Zeichen in Großbuchstaben enthält. Es kann nur einmal verwendet werden und ist für einen Zeitraum von 20 Minuten gültig. Wenn ein PassTicket von einem lokalen RACF-System generiert wird, überprüft RACF nur, ob das Profil vorhanden ist, nicht jedoch, ob der Benutzer über eine Berechtigung für das Profil verfügt. Wenn das PassTicket auf einem fernen System generiert wurde, prüft RACF den Zugriff der Benutzer-ID auf das Profil. Die vollständigen Informationen zu PassTickets finden Sie im Handbuch *z/OS SecureWay Security Server RACF Security Administrator's Guide*.

PassTickets in IMS-Headern werden von IBM MQan RACF vergeben, nicht IMS.

z/OS Einen z/OS -Warteschlangenmanager zur Sicherheit mit gemischter Groß-/Kleinschreibung migrieren

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager zur Sicherheit mit gemischter Groß-/Kleinschreibung zu migrieren. Überprüfen Sie die Stufe des Sicherheitsprodukts, die Sie verwenden, und aktivieren Sie die neuen IBM MQ-Klassen für externe Sicherheitsmanager. Führen Sie den Befehl **REFRESH SECURITY** aus, um die Mixed-Case-Profile zu aktivieren.

Vorbereitende Schritte

1. Stellen Sie sicher, dass alle IBM MQ-Klassen für externe Sicherheitsmanager aktiviert sind.
2. Stellen Sie sicher, dass der WS-Manager gestartet wurde.

Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager zur Sicherheit mit gemischter Groß-/Kleinschreibung zu konvertieren.

Vorgehensweise

1. Kopieren Sie alle vorhandenen Profile und Zugriffsebenen aus den Klassen mit Großbuchstaben in die entsprechende Klasse für externe Sicherheitsmanager mit gemischter Groß-/Kleinschreibung.
   a) `MQADMIN` auf `MXADMIN`.
   b) `MQPROC` auf `MXPROC`.
   c) `MQNLIST` auf `MXNLIST`.
   d) `MQQUEUE` auf `MXQUEUE`.
2. Ändern Sie den Wert des Attributs SCYCASE des MQ-Warteschlangenmanagers in `MIXED`, indem Sie den folgenden Befehl ausgeben.

   ```
   ALTER QMGR SCYCASE(MIXED)
   ```

3. Aktivieren Sie die Sicherheitsprofile, indem Sie den folgenden Befehl ausgeben.

   ```
   REFRESH SECURITY(*) TYPE(CLASSES)
   ```

4. Testen Sie, ob Ihre Sicherheitsprofile ordnungsgemäß funktionieren.

Nächste Schritte

Überprüfen Sie Ihre Objektdefinitionen und erstellen Sie gegebenenfalls neue Profile mit gemischter Groß-/Kleinschreibung. Verwenden Sie dazu den Befehl **REFRESH SECURITY** , um die Profile zu aktivieren.

IBM MQ MQI client-Sicherheit einrichten

Sie müssen die IBM MQ MQI client-Sicherheit berücksichtigen, damit die Clientanwendungen keinen unbeschränkten Zugriff auf Ressourcen auf dem Server haben.

Wenn Sie eine Clientanwendung ausführen, führen Sie die Anwendung nicht mit einer Benutzer-ID aus, die über mehr Zugriffsberechtigungen verfügt als erforderlich, z. B. ein Benutzer in der Gruppe `mqm` oder auch der `mqm` -Benutzer selbst.

Wenn Sie eine Anwendung als Benutzer mit zu vielen Zugriffsberechtigungen ausführen, laufen Sie Gefahr, dass der Zugriff auf die Anwendung und die Änderung von Teilen des Warteschlangenmanagers durch Zufall oder böswillig erfolgt.

Es gibt zwei Aspekte der Sicherheit zwischen einer Clientanwendung und ihrem WS-Manager-Server: Authentifizierung und Zugriffssteuerung.

- Die Authentifizierung kann verwendet werden, um sicherzustellen, dass die Clientanwendung, die als bestimmter Benutzer ausgeführt wird, die Person ist, die sie angeben. Durch die Verwendung der Authentifizierung können Sie verhindern, dass ein Angreifer Zugriff auf Ihren Warteschlangenmanager erhält, indem Sie eine Ihrer Anwendungen impersonieren.

  Ab IBM MQ 8.0 wird die Authentifizierung durch eine von zwei Optionen bereitgestellt:

  – Die Verbindungsauthentifizierungsfunktion.

    Weitere Informationen zur Verbindungsauthentifizierung finden Sie unter „Verbindungsauthentifizierung” auf Seite 75.

  – Die gegenseitige Authentifizierung in TLS wird verwendet.

    Weitere Informationen zu TLS finden Sie unter „Mit SSL/TLS arbeiten” auf Seite 294.

- Die Zugriffssteuerung kann verwendet werden, um Zugriffsberechtigungen für einen bestimmten Benutzer oder eine bestimmte Gruppe von Benutzern zu erteilen oder zu entfernen. Wenn Sie eine Clientanwendung mit einem speziell erstellten Benutzer (oder einem Benutzer in einer bestimmten Gruppe) ausführen, können Sie die Zugriffssteuerungen verwenden, um sicherzustellen, dass die Anwendung nicht auf Teile Ihres Warteschlangenmanagers zugreifen kann, für die die Anwendung nicht vorgesehen ist.

  Wenn Sie die Zugriffssteuerung einrichten, müssen Sie die Kanalauthentifizierungsregeln und das MCAUSER-Feld in einem Kanal berücksichtigen. Beide Funktionen haben die Möglichkeit, die Benutzer-ID, die für die Überprüfung der Zugriffssteuerungsberechtigungen verwendet wird, zu ändern.

  Weitere Informationen zur Zugriffssteuerung finden Sie unter „Autorisieren des Zugriffs auf Objekte” auf Seite 375.

Wenn Sie eine Clientanwendung so konfiguriert haben, dass sie eine Verbindung zu einem bestimmten Kanal mit einer eingeschränkten ID herstellt, der Kanal jedoch eine Administrator-ID in ihrem MCAUSER-Feld hat, wenn die Clientanwendung erfolgreich verbunden ist, wird die Administrator-ID für den Zugriff auf Steuerprüfungen verwendet. Daher hat die Clientanwendung volle Zugriffsberechtigungen für Ihren Warteschlangenmanager.

Weitere Informationen zum Attribut MCAUSER finden Sie unter „Zuordnung einer Clientbenutzer-ID zu einer MCAUSER-Benutzer-ID” auf Seite 414.

Kanalauthentifizierungsregeln können auch als Methode für die Steuerung des Zugriffs auf einen Warteschlangenmanager verwendet werden, indem bestimmte Regeln und Kriterien für die Annahme einer Verbindung festgelegt werden.

Weitere Informationen zu Kanalauthentifizierungsregeln finden Sie unter „Kanalauthentifizierungssätze” auf Seite 54.

Angeben, dass nur FIPS-zertifizierte CipherSpecs während der Ausführung auf dem MQI-Client verwendet werden

Erstellen Sie Ihre Schlüsselrepositorys mit FIPS-konformer Software und geben Sie dann an, dass der Kanal FIPS-zertifizierte CipherSpecs verwenden muss.

Anmerkung: Unter AIX, Linux, and Windowsstellt IBM MQ Konformität mit FIPS 140-2 über das Verschlüsselungsmodul "IBM Crypto for C" bereit. Das Zertifikat für dieses Modul wurde in den Langzeitstatus versetzt. Kunden sollten das IBM Crypto for C-Zertifikat anzeigen und sich über Empfehlungen von NIST im Klaren sein. Ein Ersatz-FIPS 140-3-Modul ist derzeit in Bearbeitung und sein Status kann angezeigt werden, indem in der NIST-CMVP-Module in der Prozesslistenach ihm gesucht wird.

Um zur Laufzeit FIPS-konform zu sein, müssen die Schlüsselrepositorys mit nur FIPS-konformer Software wie runmqakm mit der Option -fips erstellt und verwaltet worden sein.

Sie können angeben, dass ein TLS-Kanal nur FIPS-zertifizierte CipherSpecs auf drei Arten verwenden muss, die in der Reihenfolge der Vorrangstellung aufgelistet sind:

1. Setzen Sie das Feld FipsRequired in der MQSCO-Struktur auf MQSSL_FIPS_YES.
2. Setzen Sie die Umgebungsvariable MQSSLFIPS auf YES.
3. Setzen Sie das Attribut "SSLFipsRequired" in der Clientkonfigurationsdatei auf YES.

FIPS-zertifizierte CipherSpecs sind standardmäßig nicht erforderlich.

Diese Werte haben die gleichen Bedeutungen wie die entsprechenden Parameterwerte in ALTER QMGR SSLFIPS (siehe ALTER QMGR). Wenn der Clientprozess derzeit keine aktiven TLS-Verbindungen hat und ein FipsRequired-Wert ordnungsgemäß in einem SSL-MQCONNX angegeben ist, müssen alle nachfolgenden TLS-Verbindungen, die diesem Prozess zugeordnet sind, nur die CipherSpecs verwenden, die diesem Wert zugeordnet sind. Dies gilt so lange, bis diese und alle anderen TLS-Verbindungen gestoppt wurden. In dieser Phase kann ein nachfolgender MQCONNX-Wert einen neuen Wert für FipsRequired bereitstellen.

Wenn Verschlüsselungshardware vorhanden ist, können die von IBM MQ verwendeten Verschlüsselungsmodule so konfiguriert werden, dass es sich dabei um die vom Hardwareprodukt bereitgestellten Module handelt, die bis zu einer bestimmten Ebene FIPS-zertifiziert sein können. Die konfigurierbaren Module und die Angabe, ob sie FIPS-zertifiziert sind, ist abhängig vom verwendeten Hardwareprodukt.

Falls möglich, wenn FIPS-only CipherSpecs konfiguriert ist, weist der MQI-Client Verbindungen zurück, die eine Nicht-FIPS-CipherSpec-Spezifikation mit MQRC_SSL_INITIALIZATION_ERROR angeben. Es kann nicht garantiert werden, dass IBM MQ alle Verbindungen dieser Art ablehnt. Es liegt in der eigenen Verantwortung des Kunden, zu ermitteln, ob die IBM MQ-Konfiguration mit FIPS kompatibel ist.

Zugehörige Konzepte

„Federal Information Processing Standards (FIPS) für AIX, Linux, and Windows” auf Seite 37
Wenn die Verschlüsselung in einem SSL/TLS-Kanal auf AIX, Linux, and Windows-Systemen erforderlich ist, verwendet IBM MQ ein Verschlüsselungspaket mit dem Namen „IBM Crypto for C (ICC)“. Auf AIX, Linux, and Windows-Plattformen erfüllt die ICC-Software das Federal Information Processing Standards (FIPS) Cryptomodule Validation Program des US National Institute of Standards and Technology (Federal Information Processing Standards) auf Ebene 140-2.

Zugehörige Verweise

FipsRequired (MQLONG)
MQSSLFIPS
SSL-Zeilengruppe der Clientkonfigurationsdatei

AIX TLS-Clientanwendungen mit mehreren Installationen von GSKit V8.0 unter AIX ausführen

Bei TLS-Client-Anwendungen aufAIX könnenMQRC_CHANNEL_CONFIG_ERROR und FehlerAMQ6175auftreten, wenn sie aufAIX Systemen mit mehreren GSKitV8.0-Installationen ausgeführt werden.

Wenn Client-Anwendungen auf einemAIX System mit mehreren GSKit V8.0-Installationen ausgeführt werden, können die Client-VerbindungsaufrufeMQRC_CHANNEL_CONFIG_ERRORzurückgeben, wenn sie TLS verwenden. Die/var/mqm/errors Protokolle zeichnen die FehlerAMQ6175 undAMQ9220 für die fehlgeschlagene Client-Anwendung auf, zum Beispiel:

```
09/08/11 11:16:13 - Process(24412.1) User(user) Program(example)
Host(machine.example.ibm.com) Installation(Installation1)
VRMF(7.1.0.0)
AMQ6175: The system could not dynamically load the shared library
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so'. The system returned
error number '8' and error message 'Symbol resolution failed
for /usr/mqm/gskit8/lib64/libgsk8ssl_64.so because:
Symbol VALUE_EC_NamedCurve_secp256r1__9GSKASNOID (number 16) is not
exported from dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_NamedCurve_secp384r1__9GSKASNOID (number 17) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_NamedCurve_secp521r1__9GSKASNOID (number 18) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecPublicKey__9GSKASNOID (number 19) is not exported from dependent
module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecdsa_with_SHA1__9GSKASNOID (number 20) is not exported from
dependent module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.
Symbol VALUE_EC_ecdsa__9GSKASNOID (number 21) is not exported from dependent
module /db2data/db2inst1/sqllib/lib64/libgsk8cms_64.so.'.

EXPLANATION:
This message applies to AIX systems. The shared library
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so' failed
to load correctly due to a problem with the library.
ACTION:
Check the file access permissions and that the file has not been corrupted.
----- amqxufnx.c : 1284 ---------------------------------------------------------
09/08/11 11:16:13 - Process(24412.1) User(user) Program(example)
Host(machine.example.ibm.com) Installation(Installation1)
VRMF(7.1.0.0)
AMQ9220: The GSKit communications program could not be loaded.

EXPLANATION:
The attempt to load the GSKit library or procedure
'/usr/mqm/gskit8/lib64/libgsk8ssl_64.so' failed with error code
536895861.
ACTION:
Either the library must be installed on the system or the environment changed
to allow the program to locate it.
----- amqcgska.c : 836 ----------------------------------------------------------
```

Eine häufige Ursache für diesen Fehler ist, dass die Einstellung der Umgebungsvariablen LIBPATH oderLD_LIBRARY_PATH dazu geführt hat, dass derIBM MQ Client einen gemischten Satz von Bibliotheken aus zwei verschiedenen GSKit V8.0-Installationen geladen hat. Die Ausführung einer IBM MQ-Clientanwendung in einer Db2-Umgebung kann diesen Fehler verursachen.

Um diesen Fehler zu vermeiden, schließen Sie die IBM MQ-Bibliotheksverzeichnisse am Anfang des Bibliothekspfads ein, damit die IBM MQ-Bibliotheken Vorrang haben. Dies kann mit dem Befehl **setmqenv** mit dem Parameter **-k** erreicht werden. Beispiel:

```
. /usr/mqm/bin/setmqenv -s -k
```

Weitere Informationen zur Verwendung des Befehls **setmqenv** finden Sie unter setmqenv (IBM MQ-Umgebung festlegen)

IBM i Kommunikation für SSL oder TLS unter IBM i einrichten

Für die sichere Kommunikation, die die verschlüsselten SSL- oder TLS-Sicherheitsprotokolle verwendet, müssen die Kommunikationskanäle eingerichtet und die digitalen Zertifikate für die Authentifizierung verwaltet werden.

Um Ihre SSL-oder TLS-Installation einzurichten, müssen Sie die Kanäle für die Verwendung von SSL oder TLS definieren. Darüber hinaus müssen Sie digitale Zertifikate erstellen und verwalten. Auf einigen Betriebssystemen können Sie die Tests mit selbst signierten Zertifikaten ausführen. Unter IBM i müssen Sie jedoch persönliche Zertifikate verwenden, die von einer lokalen Zertifizierungsstelle signiert sind.

Umfassende Informationen zum Erstellen und Verwalten von Zertifikaten finden Sie im Abschnitt „Mit SSL/TLS unter IBM i arbeiten" auf Seite 294.

Diese Themensammlung enthält einige der Aufgaben, die an der Konfiguration der SSL-oder TLS-Kommunikation beteiligt sind, und stellt schrittweise Anleitungen zur Ausführung dieser Tasks bereit.

Sie können auch die SSL-oder TLS-Clientauthentifizierung testen, die optionale Teile der SSL-und TLS-Protokolle sind. Während des SSL-oder TLS-Handshakes ruft der SSL-oder TLS-Client immer ein digitales Zertifikat vom Server ab und validiert es. Bei der Implementierung von IBM MQ fordert der SSL- oder TLS-Server immer ein Zertifikat vom Client an.

Auf IBM i sendet der SSL- oder TLS-Client ein Zertifikat nur dann, wenn es im richtigen IBM MQ-Format gekennzeichnet ist:

- Für einen Warteschlangenmanager ist dies `ibmwebspheremq` gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinschreibung. Beispiel: `ibmwebspheremqqm1` für `QM1`.
- Für einen IBM MQ-C-Client für IBM i ist dies `ibmwebspheremq` gefolgt von Ihrer Anmeldebenutzer-ID in Kleinschreibung, z. B. `ibmwebspheremqmyuserid`.

IBM MQ verwendet bei Bezeichnungen das Präfix `ibmwebspheremq`, um eine Verwechslung mit Zertifikaten für andere Produkte zu vermeiden. Stellen Sie sicher, dass Sie die gesamte Zertifikatsbezeichnung in Kleinbuchstaben angeben.

Der SSL-oder TLS-Server überprüft das Clientzertifikat immer, wenn ein Zertifikat gesendet wird. Wenn der SSL-oder TLS-Client kein Zertifikat sendet, schlägt die Authentifizierung nur fehl, wenn das Ende des Kanals, der als SSL-oder TLS-Server fungiert, entweder mit dem Parameter SSLCAUTH oder einem Parameterwertsatz SSLPEER definiert ist. Weitere Informationen finden Sie unter Zwei WS-Manager mit SSL oder TLS verbinden .

ALW Kommunikation für SSL oder TLS unter AIX, Linux, and Windows einrichten

Für die sichere Kommunikation, die die verschlüsselten SSL- oder TLS-Sicherheitsprotokolle verwendet, müssen die Kommunikationskanäle eingerichtet und die digitalen Zertifikate für die Authentifizierung verwaltet werden.

Um Ihre SSL-oder TLS-Installation einzurichten, müssen Sie die Kanäle für die Verwendung von SSL oder TLS definieren. Darüber hinaus müssen Sie digitale Zertifikate erstellen und verwalten. Auf AIX, Linux, and Windows-Systemen können Sie die Tests mit selbst signierten Zertifikaten durchführen.

Achtung: Es ist nicht möglich, eine Kombination aus Zertifikaten, die mit Elliptic Curve und RSA signiert wurden, auf Warteschlangenmanagern zu verwenden, die mithilfe von TLS-fähigen Kanälen miteinander verknüpft werden sollen.

Alle Warteschlangenmanager, die TLS-fähige Kanäle verwenden, müssen entweder mit RSA signierte Zertifikate oder mit EC signierte Zertifikate verwenden und nicht eine Kombination aus beiden.

Weitere Informationen finden Sie unter „Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ" auf Seite 49.

Selbst signierte Zertifikate können nicht widerrufen werden, was einem Angreifer die Identität einer Identität ermöglichen könnte, nachdem ein privater Schlüssel kompromittiert wurde. CAs können ein kompromitveres Zertifikat widerrufen, das seine weitere Verwendung verhindert. CA-signierte Zertifikate sind daher sicherer in einer Produktionsumgebung zu verwenden, obwohl selbst signierte Zertifikate für ein Testsystem komfortabler sind.

Umfassende Informationen zum Erstellen und Verwalten von Zertifikaten finden Sie im Abschnitt „Mit SSL/TLS unter AIX, Linux, and Windows arbeiten" auf Seite 307.

Diese Themensammlung enthält einige der Tasks, die an der Einrichtung der SSL-Kommunikation beteiligt sind, und stellt schrittweise Anleitungen zur Ausführung dieser Tasks bereit.

Sie können auch die SSL-oder TLS-Clientauthentifizierung testen, die ein optionaler Teil der Protokolle ist. Während des SSL-oder TLS-Handshakes ruft der SSL-oder TLS-Client immer ein digitales Zertifikat vom Server ab und validiert es. Bei der Implementierung von IBM MQ fordert der SSL- oder TLS-Server immer ein Zertifikat vom Client an.

Unter AIX, Linux, and Windows sendet der SSL- oder TLS-Client ein Zertifikat nur dann, wenn es im richtigen IBM MQ-Format gekennzeichnet ist:

- Für einen Warteschlangenmanager gilt das Format `ibmwebspheremq` gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinschreibung. Beispiel für `QM1`: `ibmwebspheremqqm1`
- Für einen IBM MQ-Client ist dies `ibmwebspheremq` gefolgt von Ihrer Anmeldebenutzer-ID in Kleinschreibung, z. B. `ibmwebspheremqmyuserid`.

IBM MQ verwendet bei Bezeichnungen das Präfix `ibmwebspheremq`, um eine Verwechslung mit Zertifikaten für andere Produkte zu vermeiden. Stellen Sie sicher, dass Sie die gesamte Zertifikatsbezeichnung in Kleinbuchstaben angeben.

Der SSL-oder TLS-Server überprüft das Clientzertifikat immer, wenn ein Zertifikat gesendet wird. Wenn der Client kein Zertifikat sendet, schlägt die Authentifizierung nur fehl, wenn das Ende des Kanals, der als SSL-oder TLS-Server fungiert, entweder mit dem Parameter SSLCAUTH oder einem Parameterwertsatz SSLPEER definiert ist. Weitere Informationen finden Sie unter Zwei WS-Manager mit SSL oder TLS verbinden .

z/OS Kommunikation für SSL oder TLS unter z/OS einrichten

Für die sichere Kommunikation, die die verschlüsselten SSL- oder TLS-Sicherheitsprotokolle verwendet, müssen die Kommunikationskanäle eingerichtet und die digitalen Zertifikate für die Authentifizierung verwaltet werden.

Um Ihre SSL-oder TLS-Installation einzurichten, müssen Sie die Kanäle für die Verwendung von SSL oder TLS definieren. Darüber hinaus müssen Sie digitale Zertifikate erstellen und verwalten. Unter z/OS können Sie die Tests mit selbst signierten Zertifikaten oder mit persönlichen Zertifikaten ausführen, die von einer lokalen Zertifizierungsstelle (CA) signiert wurden.

Selbst signierte Zertifikate können nicht widerrufen werden, was einem Angreifer die Identität einer Identität ermöglichen könnte, nachdem ein privater Schlüssel kompromittiert wurde. CAs können ein kompromitveres Zertifikat widerrufen, das seine weitere Verwendung verhindert. CA-signierte Zertifikate sind daher sicherer in einer Produktionsumgebung zu verwenden, obwohl selbst signierte Zertifikate für ein Testsystem komfortabler sind.

Umfassende Informationen zum Erstellen und Verwalten von Zertifikaten finden Sie im Abschnitt „Mit SSL/TLS unter z/OS arbeiten" auf Seite 341.

Weitere Informationen finden Sie unter den Parametern CERTLABL und CERTQSGL des Befehls ALTER QMGR und unter dem Parameter CERLABL des Befehls DEFINE CHANNEL .

Im Folgenden ist die Reihenfolge für die Vorrangstellung dargestellt:

- Channel CERTLABL, Parameter
- QMGR CERTQSGL-Parameter, wenn der Kanal gemeinsam genutzt wird.

Bei einem Senderkanal bedeutet dies, dass die Übertragungswarteschlange (XMITQ) gemeinsam genutzt wird. Bei einem Empfängerkanal bedeutet dies, dass der Kanal über das gemeinsam genutzte Empfangsprogramm gestartet wurde, d. h. das Empfangsprogramm mit INDISP (GROUP).

- QMGR-CERTLABL
- Die Standardbezeichnung `ibmWebSphereMQ` gefolgt vom Namen der Gruppe mit gemeinsamer Warteschlange für gemeinsame Kanäle oder dem Namen des Warteschlangenmanagers.

Diese Themensammlung enthält einige der Aufgaben, die an der Konfiguration der SSL-oder TLS-Kommunikation beteiligt sind, und stellt schrittweise Anleitungen zur Ausführung dieser Tasks bereit.

Sie können auch die SSL-oder TLS-Clientauthentifizierung testen, die ein optionaler Teil der Protokolle ist. Während des SSL-oder TLS-Handshakes ruft der SSL-oder TLS-Client immer ein digitales Zertifikat vom Server ab und validiert es. Bei der Implementierung von IBM MQ fordert der SSL- oder TLS-Server immer ein Zertifikat vom Client an.

Wenn der Kanal gemeinsam genutzt wird, versucht der Kanal zuerst, ein Zertifikat für die Gruppe mit gemeinsamer Warteschlange zu finden. Wenn kein Zertifikat für eine Gruppe mit gemeinsamer Warteschlange gefunden wird, wird versucht, ein Zertifikat für den Warteschlangenmanager zu finden.

Unterz/OS verwendet IBM MQ in einer Bezeichnung das Präfix `ibmWebSphereMQ`, um eine Verwechslung mit Zertifikaten für andere Produkte zu vermeiden.

Der SSL-oder TLS-Server überprüft das Clientzertifikat immer, wenn ein Zertifikat gesendet wird. Wenn der SSL-oder TLS-Client kein Zertifikat sendet, schlägt die Authentifizierung nur fehl, wenn das Ende des Kanals, der als SSL-oder TLS-Server fungiert, entweder mit dem Parameter SSLCAUTH oder einem Parameterwertsatz SSLPEER definiert ist. Weitere Informationen finden Sie unter Zwei WS-Manager mit SSL oder TLS verbinden .

Mit SSL/TLS arbeiten

In diesen Abschnitten finden Sie Anweisungen zum Ausführen von einzelnen Tasks im Zusammenhang mit der Verwendung von TLS mit IBM MQ.

Viele von ihnen werden als Schritte in den in den folgenden Abschnitten beschriebenen Tasks der höheren Ebene verwendet:

- „Benutzer identifizieren und authentifizieren” auf Seite 354
- „Autorisieren des Zugriffs auf Objekte” auf Seite 375
- „Vertraulichkeit von Nachrichten” auf Seite 447
- „Datenintegrität von Nachrichten” auf Seite 509
- „Cluster sicher halten” auf Seite 510

IBM i Mit SSL/TLS unter IBM i arbeiten

Diese Themensammlung enthält Anweisungen für einzelne Tasks, die mit Transport Layer Security (TLS) in IBM MQ for IBM i arbeiten.

Für IBM i ist die TLS-Unterstützung ein integraler Bestandteil des Betriebssystems. Stellen Sie sicher, dass die in den Hardware- und Softwarevoraussetzungen unter IBM i aufgeführten Voraussetzungen installiert sind.

Unter IBM i verwalten Sie Schlüssel und digitale Zertifikate mit dem Tool Digital Certificate Manager (DCM).

Zugriff auf DCM

Beachten Sie diese Anweisungen für den Zugriff auf die DCM-Schnittstelle.

Informationen zu diesem Vorgang

Führen Sie in einem Web-Browser mit Rahmenunterstützung die folgenden Schritte aus:

Vorgehensweise

1. Wechseln Sie zu `http://`*machine*`.domain:2001` oder `https://`*machine*`.domain:2010`, wobei *machine* für den Namen Ihres Computers steht.
2. Geben Sie bei der entsprechenden Aufforderung ein gültiges Benutzerprofil und Kennwort ein.

   Vergewissern Sie sich, dass Ihr Benutzerprofil über die Sonderberechtigungen *ALLOBJ und *SECADM verfügt, damit Sie neue Zertifikatsspeicher erstellen können. Wenn Sie nicht über diese Sonderberechtigungen verfügen, können Sie lediglich Ihre persönlichen Zertifikate verwalten und die Objektsignaturen für Objekte anzeigen, für die Sie eine Berechtigung haben. Wenn Sie zur Verwendung einer Anwendung für Objektsignaturen berechtigt sind, können Sie über DCM auch Objekte signieren.
3. Klicken Sie auf der Seite 'Internet Configurations' auf **Digital Certificate Manager**.

   Die Seite 'Digital Certificate Manager' wird aufgerufen.

Zertifikat unter IBM i einem Warteschlangenmanager zuordnen

Verwenden Sie DCM, um einem Warteschlangenmanager ein Zertifikat zuzuordnen.

Verwenden Sie die konventionelle Verwaltung digitaler Zertifikate von IBM i, um ein Zertifikat einem Warteschlangenmanager zuzuordnen. Dies bedeutet, dass Sie angeben können, dass ein Warteschlangenmanager den Systemzertifikatsspeicher verwendet, und dass der Warteschlangenmanager für die Verwendung als Anwendung mit Digital Certificate Manager registriert ist. Ändern Sie dazu den Wert des Attributs des Warteschlangenmanagers **SSLKEYR** in *SYSTEM .

Wenn der Parameter **SSLKEYR** in *SYSTEMgeändert wird, registriert IBM MQ den Warteschlangenmanager als Serveranwendung mit der eindeutigen Anwendungsbezeichnung QIBM_WEBSPHERE_MQ_QMGRNAME und einer Bezeichnung mit der Beschreibung Qmgrname (WMQ). Beachten Sie, dass die Attribute des Kanals **CERTLABL** nicht verwendet werden, wenn Sie den Zertifikatsspeicher *SYSTEM verwenden. Der WS-Manager wird dann als Serveranwendung in Digital Certificate Manager angezeigt, und Sie können dieser Anwendung alle Server-oder Clientzertifikate im Systemspeicher zuordnen.

Da der Warteschlangenmanager als Anwendung registriert ist, können erweiterte Funktionen von DCM, wie z. B. die Definition von CA-Anerkennungslisten, ausgeführt werden.

Wenn der Parameter **SSLKEYR** in einen anderen Wert als *SYSTEMgeändert wird, nimmt IBM MQ die Registrierung des Warteschlangenmanagers als Anwendung mit Digital Certificate Manager. Wenn ein WS-Manager gelöscht wird, wird er auch von DCM zurückgenommen. Benutzer, die über die erforderliche *SECADM-Berechtigung verfügen, können Anwendungen manuell in DCM registrieren bzw. daraus entfernen.

Schlüsselrepository unter IBM i einrichten

Ein Schlüsselrepository muss an beiden Enden der Verbindung konfiguriert werden. Die Standardzertifikatsspeicher können verwendet werden, oder Sie können eigene Zertifikate erstellen.

Für eine TLS-Verbindung ist an jedem Ende der Verbindung ein *Schlüsselrepository* erforderlich. Jeder Warteschlangenmanager und jeder IBM MQ MQI client muss auf ein Schlüsselrepository zugreifen können. Wenn Sie mit einem Dateinamen und einem Kennwort auf das Schlüsselrepository zugreifen möchten (d. a. nicht mit der Option *SYSTEM), stellen Sie sicher, dass das Benutzerprofil QMQM die folgenden Berechtigungen hat:

- Die Berechtigung für das Verzeichnis ausführen, das das Schlüsselrepository enthält.
- Leseberechtigung für die Datei, die das Schlüsselrepository enthält

Weitere Informationen finden Sie unter „Das SSL/TLS-Schlüsselrepository” auf Seite 27. Beachten Sie, dass die Kanalattribute **CERTLABL** nicht verwendet werden, wenn Sie den Zertifikatsspeicher *SYSTEM verwenden.

Unter IBM i werden digitale Zertifikate in einem Zertifikatsspeicher gespeichert, der mit DCM verwaltet wird. Diese digitalen Zertifikate verfügen über Bezeichnungen, mit denen ein Zertifikat einem Warteschlangenmanager oder einem IBM MQ MQI client zugeordnet wird. TLS verwendet die Zertifikate zu Authentifizierungszwecken.

Die Bezeichnung ist der Wert des Attributs **CERTLABL**, wenn dieses festgelegt ist, oder der Standardwert `ibmwebspheremq`, an den der Name des Warteschlangenmanagers oder die Anmelde-ID des IBM MQ MQI client-Benutzers in Kleinschreibung angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

Der Name des Zertifikatsspeichers für den Warteschlangenmanager oder den IBM MQ MQI client enthält einen Pfad und einen Stammnamen. Der Standardpfad ist `/QIBM/UserData/ICSS/Cert/Server/` , und der Standardstammname ist `Default`. Unter IBM iwird der Standardzertifikatsspeicher `/QIBM/UserData/ICSS/Cert/Server/Default.kdb`auch als `*SYSTEM`bezeichnet. Optional können Sie Ihren eigenen Pfad und Stammnamen definieren.

Wenn Sie einen eigenen Pfad oder Dateinamen definieren, legen Sie die Berechtigungen für die Datei fest, um den Zugriff auf diese Datei genau zu steuern.

„Position des Schlüsselrepositorys für einen Warteschlangenmanager unter IBM i ändern” auf Seite 297 enthält Informationen zur Angabe des Namens des Zertifikatsspeichers. Sie können den Namen des Zertifikatsspeichers vor oder nach dem Erstellen des Zertifikatsspeichers angeben.

Anmerkung: Die Operationen, die Sie mit DCM ausführen können, werden möglicherweise von der Berechtigung Ihres Benutzerprofils begrenzt. Sie benötigen z. B. die Berechtigungen *ALLOBJ und *SECADM, um ein CA-Zertifikat zu erstellen.

Zertifikatsspeicher unter IBM i erstellen
Wenn Sie den Standardzertifikatsspeicher nicht verwenden möchten, führen Sie diese Prozedur aus, um eigene Zertifikate zu erstellen.

Informationen zu diesem Vorgang

Erstellen Sie nur dann einen neuen Zertifikatsspeicher, wenn Sie nicht den Standardzertifikatsspeicher von IBM i verwenden möchten.

Um anzugeben, dass der Zertifikatsspeicher des IBM i -Systems verwendet werden soll, ändern Sie den Wert des Attributs SSLKEYR des Warteschlangenmanagers in *SYSTEM. Dieser Wert gibt an, dass der Warteschlangenmanager den Systemzertifikatsspeicher verwendet und der Warteschlangenmanager für die Verwendung als eine Anwendung mit Digital Certificate Manager (DCM) registriert ist.

Vorgehensweise

1. Greifen Sie auf die DCM-Schnittstelle zu, wie in „Zugriff auf DCM” auf Seite 294 beschrieben.
2. Klicken Sie in der Navigationsanzeige auf **Neuen Zertifikatsspeicher erstellen** .

   Die Seite Create New Certificate Store (Neue Zertifikatsspeicher erstellen) wird im Taskrahmen angezeigt.
3. Wählen Sie im Taskrahmen **Other System Certificate Store** aus und klicken Sie auf **Continue** .

   Die Seite 'Zertifikat in neuem Zertifikatsspeicher erstellen' wird im Taskrahmen angezeigt.
4. Wählen Sie **No-Do not create a certificate in the certificate store** aus und klicken Sie auf **Continue** .

   Die Seite Zertifikatsspeichername und Kennwort wird im Taskrahmen angezeigt.
5. Geben Sie im Feld **Pfad und Dateiname des Zertifikatsspeichers** einen IFS-Pfad und Dateinamen ein, z. B. `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`.
6. Geben Sie ein Kennwort in das Feld **Kennwort** ein, und geben Sie es erneut in das Feld **Kennwort bestätigen** ein. Klicken Sie auf **Weiter** .

   Notieren Sie sich das Kennwort (die Groß-/Kleinschreibung muss beachtet werden), da Sie es benötigen, wenn Sie den Repositoryschlüssel verstellen.
7. Schließen Sie das Browserfenster, um DCM zu verlassen.

Nächste Schritte

Wenn Sie den Zertifikatsspeicher mit DCM erstellt haben, stellen Sie sicher, dass Sie das Kennwort verlegen, wie im Abschnitt „Kennwort für Zertifikatsspeicher auf IBM i-Systemen speichern” auf Seite 297 beschrieben.

Zugehörige Tasks

„Zertifikat in ein Schlüsselrepository unter IBM i importieren” auf Seite 302
Gehen Sie wie folgt vor, um ein Zertifikat zu importieren.

Kennwort für Zertifikatsspeicher auf IBM i-Systemen speichern

Speichern Sie das Kennwort des Zertifikatsspeichers mit Hilfe von CL-Befehlen.

Die folgenden Anweisungen gelten für das Speichern des Zertifikatsspeicherkennworts unter IBM i für einen Warteschlangenmanager. Wenn Sie bei einem IBM MQ MQI clientnicht den Zertifikatsspeicher *SYSTEM verwenden (d. h., die MQSSLKEYR-Umgebung ist auf einen anderen Wert als *SYSTEM gesetzt), können Sie alternativ die im Abschnitt „Kennwort für Zertifikatsspeicher speichern” auf Seite 306 unter „IBM MQ-SSL-Clientdienstprogramm (amqrsslc) für IBM i” auf Seite 305beschriebene Prozedur ausführen.

Wenn Sie angegeben haben, dass der Zertifikatsspeicher *SYSTEM verwendet werden soll (indem Sie den Wert des Attributs SSLKEYR des Warteschlangenmanagers in *SYSTEM ändern), müssen Sie die folgenden Schritte nicht ausführen.

Wenn Sie den Zertifikatsspeicher mit DCM erstellt haben, verwenden Sie die folgenden Befehle, um das Kennwort zu verlegen:

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

Bei dem Kennwort muss die Groß-/Kleinschreibung beachtet werden Er muss in einfache Anführungszeichen eingegeben werden, wie Sie ihn in Schritt 6 von „Zertifikatsspeicher unter IBM i erstellen” auf Seite 296 eingegeben haben.

Anmerkung: Wenn Sie den Standardzertifikatsspeicher nicht verwenden und das Kennwort nicht verlegen, schlagen Versuche zum Starten von TLS-Kanälen fehl, da sie das für den Zugriff auf den Zertifikatsspeicher erforderliche Kennwort nicht abrufen können.

Schlüsselrepository für einen Warteschlangenmanager unter IBM i ermitteln

Verwenden Sie diese Prozedur, um die Position des Zertifikatsspeichers Ihres WS-Managers abzurufen.

Vorgehensweise

1. Zeigen Sie die Attribute des Warteschlangenmanagers mit dem folgenden Befehl an:

   ```
   DSPMQM MQMNAME('queue manager name')
   ```

2. Untersuchen Sie die Befehlsausgabe für den Pfad und den Stammnamen des Zertifikatsspeichers.
   Beispiel: `/QIBM/UserData/ICSS/Cert/Server/Default`. Hierbei steht `/QIBM/UserData/ICSS/Cert/Server` für den Pfad und `Default` für den Stammnamen.

Position des Schlüsselrepositorys für einen Warteschlangenmanager unter IBM i ändern

Ändern Sie die Position des Zertifikatsspeichers Ihres WS-Managers mit dem Befehl CHGMQM oder ALTER QMGR.

Vorgehensweise

Verwenden Sie entweder den Befehl CHGMQM oder den MQSC-Befehl ALTER QMGR, um das Schlüsselrepository-Attribut des WS-Managers festzulegen.

a) CHGMQM verwenden: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey')`

b) Verwenden von ALTER QMGR: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey')`

In beiden Fällen hat der Zertifikatsspeicher den vollständig qualifizierten Dateinamen: `/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb`

Nächste Schritte

Wenn Sie die Position des Zertifikatsspeichers eines WS-Managers ändern, werden die Zertifikate nicht von der alten Position übertragen. Wenn die CA-Zertifikate, die bei der Erstellung des Zertifikatsspeichers vorinstalliert sind, nicht ausreichen, müssen Sie den neuen Zertifikatsspeicher mit Zertifikaten füllen, wie in „Zertifikat in ein Schlüsselrepository unter IBM i importieren” auf Seite 302 beschrieben. Sie müssen außerdem das Kennwort für die neue Position nach der Beschreibung im Abschnitt „Kennwort für Zertifikatsspeicher auf IBM i-Systemen speichern” auf Seite 297 verstaschen.

Zertifizierungsstelle und Zertifikat für Tests unter IBM i erstellen

Verwenden Sie diese Prozedur, um ein lokales CA-Zertifikat zu erstellen, um Zertifikatsanforderungen zu signieren und das CA-Zertifikat zu erstellen und zu installieren.

Vorbereitende Schritte

Die Anweisungen in diesem Abschnitt gehen davon aus, dass eine lokale Zertifizierungsstelle (CA) nicht vorhanden ist. Wenn eine lokale Zertifizierungsinstanz vorhanden ist, fahren Sie mit dem Abschnitt „Serverzertifikat unter IBM i anfordern” auf Seite 299 weiter.

Informationen zu diesem Vorgang

Die CA-Zertifikate, die bei der Installation von TLS zur Verfügung gestellt werden, werden von der ausstellenden Zertifizierungsstelle signiert. Unter IBM i können Sie eine lokale Zertifizierungsstelle generieren, die Serverzertifikate signieren kann, um die TLS-Kommunikation auf Ihrem System zu testen. Führen Sie die folgenden Schritte in einem Webbrowser aus, um ein lokales CA-Zertifikat zu erstellen:

Vorgehensweise

1. Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
2. Klicken Sie in der Navigationsanzeige auf **Create a Certificate Authority** (Zertifizierungsstelle erstellen).

   Die Seite "Zertifizierungsstelle erstellen" wird im Taskrahmen angezeigt.
3. Geben Sie ein Kennwort in das Feld **Certificate store password** ein, und geben Sie es erneut in das Feld **Confirm password** ein.
4. Geben Sie einen Namen in das Feld **Certificate Authority (CA) name** ein, z. B. `TLS Test Certificate Authority` .
5. Geben Sie die entsprechenden Werte in die Felder **Allgemeiner Name** und **Organisation** ein, und wählen Sie ein Land aus. Geben Sie für die verbleibenden optionalen Felder die Werte ein, die Sie benötigen.
6. Geben Sie im Feld **Gültigkeitszeitraum** einen Gültigkeitszeitraum für die lokale Zertifizierungsinstanz ein.

   Der Standardwert ist 1095 Tage.
7. Klicken Sie auf **Weiter** .

   Die Zertifizierungsstelle wird erstellt, und DCM erstellt einen Zertifikatsspeicher und ein CA-Zertifikat für Ihre lokale Zertifizierungsinstanz.
8. Klicken Sie auf **Zertifikat installieren** .

   Daraufhin wird das Dialogfenster zum Download-Manager angezeigt.
9. Geben Sie den vollständigen Pfadnamen für die temporäre Datei ein, in der das CA-Zertifikat gespeichert werden soll, und klicken Sie auf **Speichern** .

10. Wenn der Download abgeschlossen ist, klicken Sie auf **Öffnen** .
    Das Fenster Zertifikat wird angezeigt.
11. Klicken Sie auf **Zertifikat installieren** .
    Der Assistent 'Zertifikatsimport' wird angezeigt.
12. Klicken Sie auf **Weiter**.
13. Wählen Sie **Zertifikatsspeicher basierend auf dem Typ des Zertifikats automatisch auswählen** aus und klicken Sie auf **Weiter** .
14. Klicken Sie auf **Fertigstellen**.
    Ein Bestätigungsfenster wird angezeigt.
15. Klicken Sie auf **OK**.
16. Klicken Sie im Fenster "Zertifikat" auf **OK** .
17. Klicken Sie auf **Weiter** .
    Die Seite "Richtlinie für Zertifizierungsstelle" wird im Taskrahmen angezeigt.
18. Wählen Sie im Feld **Erstellung von Benutzerzertifikaten zulassen** die Option **Ja** aus.
19. Geben Sie im Feld **Gültigkeitszeitraum** den Gültigkeitszeitraum der Zertifikate ein, die von Ihrer lokalen Zertifizierungsinstanz ausgestellt werden.
    Der Standardwert ist 365 Tage.
20. Klicken Sie auf **Weiter** .
    Die Seite 'Zertifikat in neuem Zertifikatsspeicher erstellen' wird im Taskrahmen angezeigt.
21. Stellen Sie sicher, dass keine der Anwendungen ausgewählt ist.
22. Klicken Sie auf **Weiter** , um die Konfiguration der lokalen Zertifizierungsinstanz abzuschließen.

Serverzertifikat unter IBM i anfordern

Digitale Zertifikate schützen vor der Nachahmung; sie zertifizieren, dass ein öffentlicher Schlüssel zu einer bestimmten Entität gehört. Ein neues Serverzertifikat kann von einer Zertifizierungsstelle unter Verwendung des Digital Certificate Manager (DCM) angefordert werden.

Informationen zu diesem Vorgang

Führen Sie in einem Web-Browser folgende Schritte aus:

Vorgehensweise

1. Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
2. Klicken Sie in der Navigationsanzeige auf **Select a Certificate Store** (Zertifikatsspeicher auswählen).
   Die Seite 'Select a Certificate Store' (Zertifikatsspeicher auswählen) wird im Taskrahmen angezeigt.
3. Wählen Sie den Zertifikatsspeicher aus, den Sie verwenden möchten, und klicken Sie auf **Weiter** .
4. Optional: Wenn Sie in Schritt 3 ***SYSTEM** ausgewählt haben, geben Sie das Systemspeicherkennwort ein und klicken Sie auf **Weiter** .
5. Optional: Wenn Sie **Other System Certificate Store** in Schritt 3 ausgewählt haben, geben Sie im Feld **Pfad und Dateiname des Zertifikatsspeichers** den IFS-Pfad und den Dateinamen ein, den Sie bei der Erstellung des Zertifikatsspeichers festgelegt haben. Geben Sie auch ein Kennwort in das Feld **Certificate Store Password** ein. Klicken Sie anschließend auf **Weiter** .
6. Klicken Sie in der Navigationsanzeige auf **Zertifikat erstellen** .
7. Wählen Sie im Taskrahmen den Radioknopf **Server-oder Clientzertifikat** aus und klicken Sie auf **Weiter** .
   Die Seite "Select a Certificate Authority (CA)" wird im Taskrahmen angezeigt.
8. Wenn Sie eine lokale Zertifizierungsinstanz auf Ihrer Workstation haben, wählen Sie entweder die lokale Zertifizierungsinstanz oder eine kommerzielle CA aus, um das Zertifikat zu signieren. Wählen Sie das Optionsfeld für die gewünschte Zertifizierungsstelle aus und klicken Sie auf **Weiter** .
   Die Seite "Zertifikat erstellen" wird im Taskrahmen angezeigt.

9. Optional: Geben Sie für einen Warteschlangenmanager in das Feld **Zertifikatsbezeichnung** die Zertifikatsbezeichnung ein.

   Der Kennsatz ist entweder der Wert des Attributs **CERTLABL** , wenn er festgelegt ist, oder der Standardwert `ibmwebspheremq` mit dem Namen des angehängten Warteschlangenmanagers in Kleinbuchstaben. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

   Geben Sie beispielsweise für Warteschlangenmanager QM1 den Wert `ibmwebspheremqqqm1` ein, um den Standardwert zu verwenden.
10. Optional: Geben Sie für einen IBM MQ MQI client im Feld **Certificate label** (Zertifikatsbezeichnung) den Wert `ibmwebspheremq` gefolgt von Ihrer Anmeldebenutzer-ID in Kleinschreibung ein. Beispiel: `ibmwebspheremqmyuserID`
11. Geben Sie die entsprechenden Werte in die Felder **Allgemeiner Name** und **Organisation** ein, und wählen Sie ein Land aus. Geben Sie für die verbleibenden optionalen Felder die Werte ein, die Sie benötigen.

Ergebnisse

Wenn Sie eine kommerzielle CA zum Signieren Ihres Zertifikats ausgewählt haben, erstellt DCM eine Zertifikatsanforderung im PEM-Format (Privacy-Enhanced Mail). Die Anforderung an die von Ihnen ausgewählte Zertifizierungsstelle weiterleiten.

Wenn Sie die lokale Zertifizierungsinstanz ausgewählt haben, um Ihr Zertifikat zu signieren, informiert DCM Sie darüber, dass das Zertifikat im Zertifikatsspeicher erstellt wurde und verwendet werden kann.

Serverzertifikat für IBM Key Manager unter IBM i anfordern

Gehen Sie wie hier beschrieben vor, um ein Zertifikat zu erstellen, das von Ihrer lokalen Zertifizierungsstelle (CA) signiert ist, oder um ein Serverzertifikat anzuwenden, das von einer kommerziellen Zertifizierungsstelle für den Import in das IBM-Dienstprogramm Key Management (iKeyman) signiert ist.

Informationen zu diesem Vorgang

Ein Benutzerzertifikat muss verwendet werden, wenn der Digital Certificate Manager (DCM) als Zertifikatsmanager für IBM MQ auf mehreren Plattformen verwendet wird. Führen Sie für persönliche Zertifikate, die auf andere Plattformen verteilt sind, sowie für den Import in das Dienstprogramm 'iKeyman' in einem Web-Browser die folgenden Schritte aus:

Vorgehensweise

1. Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
2. Klicken Sie im Teilfenster **Navigation** auf **Zertifikat erstellen** .

   Die Seite **Create Certificate** (Zertifikat erstellen) wird im Taskrahmen angezeigt.
3. Wählen Sie in der Anzeige **Zertifikat erstellen** das Optionsfeld **Benutzerzertifikat** aus und klicken Sie auf **Weiter** .

   Die Seite **Create User Certificate** (Benutzerzertifikat erstellen) wird angezeigt.
4. Geben Sie in der Anzeige **Create User Certificate** (Benutzerzertifikat erstellen) die erforderlichen Felder unter Zertifikatsinformationen für **Organisationsname** , **Status** oder **Provinz** , **Land** oder **Region** ein. Geben Sie optional Werte in die Felder **Organisationseinheit** und **Ort** oder **Stadt** ein. Klicken Sie auf **Weiter** .

   Als **Common name** (Allgemeiner Name) wird automatisch die Benutzer-ID festgelegt, mit der Sie auf dem iSeries-System angemeldet sind.
5. Klicken Sie in der nächsten Anzeige **Create User Certificate** (Benutzerzertifikat erstellen) auf **Install certificate** (Zertifikat installieren) und anschließend auf **Continue** (

   Es wird eine Nachricht angezeigt, die besagt: `Ihr persönliches Zertifikat wurde installiert. Sie sollten eine Sicherungskopie dieses Zertifikats aufbewahren.`
6. Klicken Sie auf **OK**.
7. Führen Sie je nach Internetbrowser, den Sie für den Zugriff auf DCM verwendet haben, die folgenden Schritte aus:

a) Wählen Sie für Microsoft Edge die Option **Tools>Internetoptionen>Registerkarte 'Inhalt'>Schaltfläche 'Zertifikate'>Persönliche Registerkarte>** aus. Wählen Sie das Zertifikat aus und klicken Sie auf **Exportieren**.

b) Wählen Sie für Mozilla Firefox die Option **Tools>Optionen>Erweitert>Registerkarte 'Verschlüsselung'>Schaltfläche 'Zertifikate anzeigen'>Registerkarte 'Zertifikate'>** aus. Wählen Sie das Zertifikat aus und klicken Sie auf **Sicherung**. Wählen Sie den Pfad und den Dateinamen aus und klicken Sie auf **OK** .

8. Übertragen Sie das exportierte Zertifikat per FTP im Binärformat an das ferne System.
9. Fügen Sie das exportierte Zertifikat aus Schritt 7 in das Dienstprogramm iKeyman in der Schlüsseldatenbank hinzu.

   a) Wenn das Zertifikat mit Microsoft Edge gespeichert wurde, folgen Sie den Anweisungen im Abschnitt Aus einer Microsoft `.pfx` -Datei importieren.

   b) Wenn das Zertifikat mit Mozilla Firefox gespeichert wurde, verwenden Sie die Anweisungen im Abschnitt Persönlichem Zertifikat in ein Schlüsselrepository importieren .

   Stellen Sie während des Imports sicher, dass die Bezeichnung des persönlichen Zertifikats und des Unterzeichnerzertifikats entsprechend der von IBM MQ erwarteten Namenskonvention geändert werden. Die Bezeichnung muss entweder der Wert des Attributs IBM MQ **CERTLABL** sein, wenn er festgelegt ist, oder der Standardwert `ibmwebspheremq` mit dem Namen des Warteschlangenmanagers, der angehängt wird, und zwar alle in Kleinschreibung. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

Serverzertifikate unter IBM i einem Schlüsselrepository hinzufügen

Gehen Sie wie folgt vor, um ein angeforderes Zertifikat zum Schlüsselrepository hinzuzufügen.

Informationen zu diesem Vorgang

Nachdem die CA Ihnen ein neues Serverzertifikat gesendet hat, fügen Sie es dem Zertifikatsspeicher hinzu, von dem Sie die Anforderung generiert haben. Wenn die Zertifizierungsstelle das Zertifikat als Teil einer E-Mail-Nachricht sendet, kopieren Sie das Zertifikat in eine separate Datei.

Anmerkung:

- Sie müssen diese Prozedur nicht ausführen, wenn das Serverzertifikat von Ihrer lokalen Zertifizierungsstelle signiert ist.
- Bevor Sie ein Serverzertifikat im PKCS#12-Format in DCM importieren, müssen Sie zunächst das entsprechende CA-Zertifikat importieren.

So importieren Sie ein Serverzertifikat in den Zertifikatsspeicher des WS-Managers:

Vorgehensweise

1. Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
2. Klicken Sie in der Taskkategorie **Zertifikate verwalten** in der Navigationsanzeige auf **Import Certificate** (Zertifikat importieren).

   Die Seite 'Import Certificate' (Zertifikat importieren) wird im Taskrahmen angezeigt.
3. Wählen Sie das Optionsfeld für Ihren Zertifikatstyp aus und klicken Sie auf **Weiter** .

   Im Taskrahmen wird entweder die Seite "Server-oder Clientzertifikat importieren" oder die Seite "Zertifizierungsstelle importieren" (CA) angezeigt.
4. Geben Sie in das Feld **Importdatei** den Dateinamen des Zertifikats ein, das Sie importieren möchten, und klicken Sie auf **Weiter** .

   DCM bestimmt automatisch das Format der Datei.
5. Wenn es sich bei dem Zertifikat um ein **Server-oder Clientzertifikat** handelt, geben Sie das Kennwort in den Taskrahmen ein und klicken Sie auf **Weiter** .

   DCM informiert Sie darüber, dass das Zertifikat importiert wurde.

Zertifikat aus einem Schlüsselrepository unter IBM i exportieren

Der Export eines Zertifikats exportiert sowohl den öffentlichen als auch den privaten Schlüssel. Diese Aktion sollte mit äußerster Vorsicht durchgeführt werden, da die Weitergabe eines privaten Schlüssels Ihre Sicherheit völlig beeinträchtigen würde.

Vorbereitende Schritte

Wenn Sie das Zertifikat eines Benutzers mit einem anderen Benutzer gemeinsam nutzen, tauschen Sie öffentliche Schlüssel aus. Dieser Prozess wird in **Aufgabe 5 beschrieben. Zertifikate gemeinsam nutzen** im Abschnitt Zertifikate gemeinsam nutzen von „Leitfaden für den Schnelleinstieg für AMS unter AIX and Linux” auf Seite 652. Wenn Sie ein Zertifikat wie hier beschrieben exportieren, exportieren Sie sowohl den öffentlichen als auch den privaten Schlüssel. Diese Aktion sollte mit äußerster Vorsicht durchgeführt werden, da die Weitergabe eines privaten Schlüssels Ihre Sicherheit völlig beeinträchtigen würde.

Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte auf dem Computer aus, aus dem das Zertifikat exportiert werden soll:

Vorgehensweise

1. Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
2. Klicken Sie in der Navigationsanzeige auf **Select a Certificate Store** (Zertifikatsspeicher auswählen).

   Die Seite 'Select a Certificate Store' (Zertifikatsspeicher auswählen) wird im Taskrahmen angezeigt.
3. Wählen Sie den Zertifikatsspeicher aus, den Sie verwenden möchten, und klicken Sie auf **Weiter** .
4. Optional: Wenn Sie in Schritt 3 ***SYSTEM** ausgewählt haben, geben Sie das Systemspeicherkennwort ein und klicken Sie auf **Weiter** .
5. Optional: Wenn Sie **Other System Certificate Store** in Schritt 3 ausgewählt haben, geben Sie im Feld **Pfad und Dateiname des Zertifikatsspeichers** den IFS-Pfad und den Dateinamen ein, den Sie bei der Erstellung Ihres Zertifikatsspeichers festgelegt haben, und geben Sie ein Kennwort in das Feld **Certificate Store Password** ein. Klicken Sie anschließend auf **Weiter** .
6. Klicken Sie in der Taskkategorie **Manage Certificates** in der Navigationsanzeige auf **Zertifikat exportieren** .

   Die Seite Export a Certificate (Zertifikat exportieren) wird im Taskrahmen angezeigt.
7. Wählen Sie das Optionsfeld für Ihren Zertifikatstyp aus und klicken Sie auf **Weiter** .

   Im Taskrahmen wird entweder die Seite "Export Server" oder "Client Certificate Authority" oder die Seite "Export Certificate Authority (CA) Certificate" angezeigt.
8. Wählen Sie das Zertifikat aus, das Sie exportieren wollen.
9. Wählen Sie das Optionsfeld aus, um anzugeben, ob das Zertifikat in eine Datei exportiert oder direkt in einen anderen Zertifikatsspeicher exportiert werden soll.
10. Wenn Sie ausgewählt haben, dass ein Server-oder Clientzertifikat in eine Datei exportiert werden soll, geben Sie die folgenden Informationen an:
    - Der Pfad und Dateiname der Position, an der das exportierte Zertifikat gespeichert werden soll.
    - Für ein persönliches Zertifikat das Kennwort, das zum Verschlüsseln des exportierten Zertifikats und des Ziel-Release verwendet wird. Für CA-Zertifikate ist die Angabe des Kennworts nicht erforderlich.
11. Wenn Sie ausgewählt haben, dass ein Zertifikat direkt in einen anderen Zertifikatsspeicher exportiert werden soll, geben Sie den Zielzertifikatsspeicher und sein Kennwort an.
12. Klicken Sie auf **Weiter** .

Zertifikat in ein Schlüsselrepository unter IBM i importieren

Gehen Sie wie folgt vor, um ein Zertifikat zu importieren.

Vorbereitende Schritte

Bevor Sie ein persönliches Zertifikat im PKCS#12-Format in DCM importieren, müssen Sie zuerst das entsprechende CA-Zertifikat importieren.

Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte auf der Maschine aus, auf die Sie das Zertifikat importieren möchten.

Vorgehensweise

1. Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
2. Klicken Sie in der Navigationsanzeige auf **Select a Certificate Store** (Zertifikatsspeicher auswählen).

   Die Seite 'Select a Certificate Store' (Zertifikatsspeicher auswählen) wird im Taskrahmen angezeigt.
3. Wählen Sie den Zertifikatsspeicher aus, den Sie verwenden möchten, und klicken Sie auf **Weiter** .
4. Optional: Wenn Sie in Schritt 3 ***SYSTEM** ausgewählt haben, geben Sie das Systemspeicherkennwort ein und klicken Sie auf **Weiter** .
5. Optional: Wenn Sie **Other System Certificate Store** in Schritt 3 ausgewählt haben, geben Sie im Feld **Pfad und Dateiname des Zertifikatsspeichers** den IFS-Pfad und den Dateinamen ein, den Sie bei der Erstellung Ihres Zertifikatsspeichers festgelegt haben, und geben Sie ein Kennwort in das Feld **Certificate Store Password** ein. Klicken Sie anschließend auf **Weiter** .
6. Klicken Sie in der Taskkategorie **Zertifikate verwalten** in der Navigationsanzeige auf **Import Certificate** (Zertifikat importieren).

   Die Seite 'Import Certificate' (Zertifikat importieren) wird im Taskrahmen angezeigt.
7. Wählen Sie das Optionsfeld für Ihren Zertifikatstyp aus und klicken Sie auf **Weiter** .

   Im Taskrahmen wird entweder die Seite "Server-oder Clientzertifikat importieren" oder die Seite "Zertifizierungsstelle importieren" (CA) angezeigt.
8. Geben Sie in das Feld **Importdatei** den Dateinamen des Zertifikats ein, das Sie importieren möchten, und klicken Sie auf **Weiter** .

   DCM bestimmt automatisch das Format der Datei.
9. Wenn es sich bei dem Zertifikat um ein **Server-oder Clientzertifikat** handelt, geben Sie das Kennwort in den Taskrahmen ein und klicken Sie auf **Weiter** . DCM informiert Sie darüber, dass das Zertifikat importiert wurde.

Zertifikate in IBM i entfernen

Verwenden Sie diese Prozedur, um persönliche Zertifikate zu entfernen.

Vorgehensweise

1. Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
2. Klicken Sie in der Navigationsanzeige auf **Select a Certificate Store** (Zertifikatsspeicher auswählen).

   Die Seite 'Select a Certificate Store' (Zertifikatsspeicher auswählen) wird im Taskrahmen angezeigt.
3. Wählen Sie das Kontrollkästchen **Other System Certificate Store** aus und klicken Sie auf **Continue** .

   Die Seite Zertifikatsspeicher und Kennwort wird angezeigt.
4. Geben Sie im Feld **Pfad und Dateiname des Zertifikatsspeichers** den IFS-Pfad und den Dateinamen ein, den Sie bei der Erstellung des Zertifikatsspeichers festgelegt haben.
5. Geben Sie ein Kennwort in das Feld **Certificate Store Password** ein. Klicken Sie auf **Weiter** .

   Die Seite Aktuelle Zertifikatsspeicher wird im Taskrahmen angezeigt.
6. Klicken Sie in der Taskkategorie **Manage Certificates** in der Navigationsanzeige auf **Zertifikat löschen** .

   Die Seite Confirm Delete Certificate (Löschen des Zertifikats bestätigen) wird im Taskrahmen angezeigt.
7. Wählen Sie das Zertifikat aus, das Sie löschen möchten. Klicken Sie auf **Löschen** .

8. Klicken Sie auf **Ja** , um zu bestätigen, dass das Zertifikat gelöscht werden soll. Klicken Sie andernfalls auf **Nein** .

   DCM informiert Sie, wenn es das Zertifikat gelöscht hat.

*Zertifikatsspeicher *SYSTEM für unidirektionale Authentifizierung unter IBM i verwenden*

Befolgen Sie diese Anweisungen, um die Einwegauthentifizierung zu konfigurieren.

Vorbereitende Schritte

- Erstellen Sie einen Warteschlangenmanager, Kanäle und Übertragungswarteschlangen.
- Erstellen Sie ein Server-oder Clientzertifikat auf dem Server-WS-Manager.
- Übertragen Sie das CA-Zertifikat an den Client-WS-Manager, und importieren Sie es in das Schlüsselrepository.
- Starten Sie einen Listener auf dem Server und den Client-WS-Managern.

Informationen zu diesem Vorgang

Um die unidirektionale Authentifizierung auf einem Computer zu verwenden, auf dem IBM i als TLS-Server ausgeführt wird, setzen Sie den Parameter SSL-Schlüsselrepository (SSLKEYR) auf *SYSTEM. Bei dieser Einstellung wird der IBM MQ-Warteschlangenmanager als Anwendung registriert. Anschließend können Sie dem WS-Manager ein Zertifikat zuordnen, um die Einmalauthentifizierung zu aktivieren.

Sie können auch private Keystores verwenden, um die Einbahnauthentifizierung zu implementieren, indem Sie ein Dummy-Zertifikat für den Client-WS-Manager im Schlüsselrepository erstellen.

Vorgehensweise

1. Führen Sie die folgenden Schritte auf dem Server und den Client-WS-Managern aus:
   a) Ändern Sie den Warteschlangenmanager, um den Parameter SSLKEYR zu setzen, indem Sie den Befehl `CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)` absetzen.
   b) Stoppt das Kennwort für das Standardschlüsselrepository, indem der Befehl `CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')` ausgegeben wird.

      Das Kennwort muss in Hochkommas angegeben werden.
   c) Ändern Sie die Kanäle so, dass die richtige CipherSpec im Parameter SSLCIPHER vorhanden ist.
   d) Aktualisieren Sie die TLS-Sicherheit, indem Sie den Befehl `RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)` ausgeben.
2. Weisen Sie das Zertifikat dem Server-WS-Manager mit DCM wie folgt zu:
   a) Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
   b) Klicken Sie in der Navigationsanzeige auf **Select a Certificate Store** (Zertifikatsspeicher auswählen).

      Die Seite 'Select a Certificate Store' (Zertifikatsspeicher auswählen) wird im Taskrahmen angezeigt.
   c) Wählen Sie den Zertifikatsspeicher *SYSTEM aus und klicken Sie auf **Weiter** .
   d) Erweitern Sie in der linken Anzeige den Eintrag **Anwendungen verwalten** .
   e) Wählen Sie die Definition **Anwendung anzeigen** aus, um zu prüfen, ob der WS-Manager als Anwendung registriert wurde.

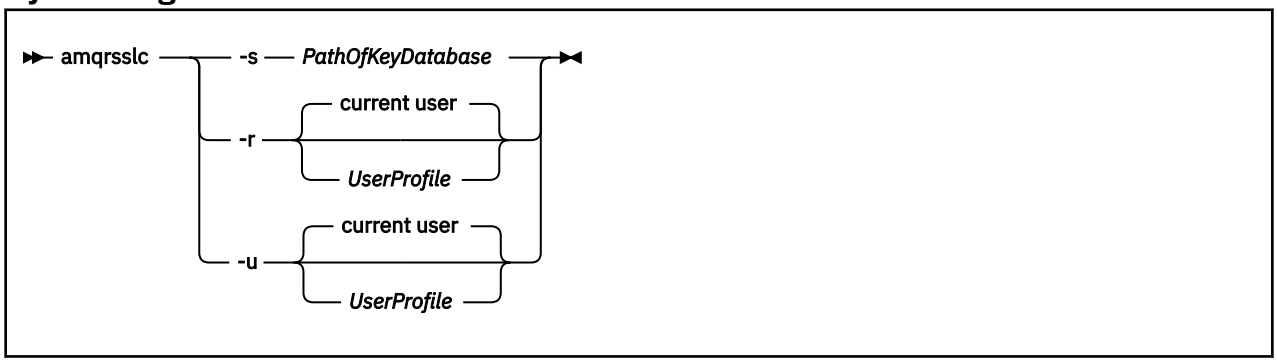
      `SSL (WMQ)` ist in der Tabelle aufgelistet.
   f) Wählen Sie **Update Certificate Assignment** aus.
   g) Wählen Sie **Server** aus und klicken Sie auf **Weiter** .
   h) Wählen Sie QMGRNAME (WMQ) aus und klicken Sie auf **Zertifikatszuordnung aktualisieren** .

i) Wählen Sie das Zertifikat aus und klicken Sie auf **Neues Zertifikat zuordnen** . Es wird ein Fenster mit dem Hinweis angezeigt, dass das Zertifikat der Anwendung zugeordnet wurde.

IBM MQ-SSL-Clientdienstprogramm (amqrsslc) für IBM i

Das IBM MQ-SSL-Clientdientprogramm (amqrsslc) für IBM i wird vom IBM MQ MQI client auf IBM i-Systemen verwendet, um die Registrierung für das Profil des Clientbenutzers vorzunehmen oder aufzuheben und das Kennwort für den Zertifikatsspeicher verdeckt zu speichern. Das Dienstprogramm kann nur von einem Benutzer mit einem Profil mit der Sonderberechtigung *ALLOBJ oder einem Member von QMQMADM ausgeführt werden, das über Optionen zum Erstellen oder Löschen von Anwendungsregistrierungen im Digital Certificate Manager (DCM) verfügt.

Syntaxdiagramm



Registrieren Sie das Clientbenutzerprofil.

Wenn der IBM MQ MQI client den Zertifikatsspeicher *SYSTEM verwendet, müssen Sie das Clientbenutzerprofil (Anmeldebenutzer) für die Verwendung als Anwendung bei Digital Certificate Manager (DCM)registrieren.

Wenn Sie das Clientbenutzerprofil registrieren möchten, führen Sie das Programm **amqrsslc** mit der Option `-r` mit *UserProfile*aus. Das beim Aufrufen von **amqrsslc** verwendete Benutzerprofil muss über die Berechtigung *USE verfügen. Wenn *UserProfile* die Option `-r` zugeordnet wird, wird *UserProfile* als Serveranwendung mit der eindeutigen Anwendungskennung QIBM_WEBSPHERE_MQ_*UserProfile* und mit einer Bezeichnung mit der Beschreibung von *UserProfile* (WMQ) registriert. Diese Serveranwendung wird dann im RZ-Modell angezeigt, und Sie können dieser Anwendung alle Server-oder Clientzertifikaten im Systemspeicher zuordnen.

Anmerkung: Wenn ein Benutzerprofil nicht mit der Option `-r` angegeben wird, wird das Benutzerprofil des Benutzers registriert, der das Tool **amqrsslc** ausführt.

Der folgende Code verwendet **amqrsslc** zum Registrieren eines Benutzerprofils. Im ersten Beispiel wird das angegebene Benutzerprofil registriert; in der zweiten ist es das Profil des angemeldeten Benutzers:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

Registrierung des Clientbenutzerprofils zurücknehmen

Um die Registrierung des Clientprofils aufzuheben, führen Sie das Programm **amqrsslc** mit der Option `-u` mit *UserProfile*aus. Das beim Aufrufen von **amqrsslc** verwendete Benutzerprofil muss über die Berechtigung *USE verfügen. Wenn Sie *UserProfile* mit der Option `-u` angeben, wird die Registrierung von *UserProfile* mit der Kennung QIBM_WEBSPHERE_MQ_*UserProfile* im DCM aufgehoben.

Anmerkung: Wenn ein Benutzerprofil nicht mit der Option `-u` angegeben wird, wird die Registrierung des Benutzerprofils des Benutzers, der das Tool **amqrsslc** ausführt, aufgehoben.

Der folgende Code verwendet **amqrsslc** , um die Registrierung eines Benutzerprofils aufzuheben. Im ersten Beispiel ist das angegebene Benutzerprofil nicht registriert, in der zweiten ist es das Profil des angemeldeten Benutzers:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

Kennwort für Zertifikatsspeicher speichern

Wenn der IBM MQ MQI client nicht den Zertifikatsspeicher *SYSTEM verwendet und keinen anderen Zertifikatsspeicher verwendet (d. h., MQSSLKEYR ist auf einen anderen Wert als *SYSTEM gesetzt), muss das Kennwort der Schlüsseldatenbank verdeckt gespeichert werden. Verwenden Sie die Option `-s` zum Stashing des Kennworts der Schlüsseldatenbank.

Im folgenden Code lautet der vollständig qualifizierte Dateiname des Zertifikatsspeichers `/Path/Of/KeyDatabase/MyKey.kdb`:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

Die Ausführung dieses Codes führt zu einer Anforderung für das Kennwort dieser Schlüsseldatenbank. Dieses Kennwort wird in einer Datei mit dem gleichen Namen wie die Schlüsseldatenbank mit der Erweiterung .sth versteckt. Diese Datei wird auf demselben Pfad wie die Schlüsseldatenbank gespeichert. Das Codebeispiel generiert eine Stashdatei von `/Path/Of/KeyDatabase/MyKey.sth`. QMQM ist der Benutzereigner und QMQMADM der Gruppeneigner für diese Datei. QMQM und QMQMADM haben Lese-, Schreibzugriff und andere Profile haben nur Leseberechtigung.

Zeitpunkt, an dem Änderungen an Zertifikaten oder dem Zertifikatsspeicher unter IBM i wirksam werden

Wenn Sie die Zertifikate in einem Zertifikatsspeicher oder in der Position des Zertifikatsspeichers ändern, werden die Änderungen in Abhängigkeit vom Typ des Kanals und der Ausführung des Kanals wirksam.

Änderungen an den Zertifikaten im Zertifikatsspeicher und an dem Schlüsselrepository-Attribut werden in den folgenden Situationen wirksam:

- Wenn ein neuer abgehender Einzelkanalprozess zuerst einen TLS-Kanal ausführt.
- Wenn ein neuer eingehender TCP/IP-Einzelkanalprozess zuerst eine Anforderung zum Starten eines TLS-Kanals empfängt.
- Wenn der MQSC-Befehl REFRESH SECURITY TYPE(SSL) zur Aktualisierung der IBM MQ-TLS-Umgebung ausgegeben wird.
- Bei Clientanwendungsprozessen, wenn die letzte TLS-Verbindung in dem Prozess geschlossen wird. Die nächste TLS-Verbindung nimmt die Zertifikatänderungen ab.
- Für Kanäle, die als Threads in einem Prozess-Pooling-Prozess (amqrmppa) ausgeführt werden, wenn der Prozess-Pooling-Prozess gestartet oder erneut gestartet wird und zuerst einen TLS-Kanal ausführt. Wenn der Prozess-Pooling-Prozess bereits einen TLS-Kanal ausgeführt hat und Sie möchten, dass die Änderung sofort wirksam wird, führen Sie den MQSC-Befehl REFRESH SECURITY TYPE (SSL) aus.
- Bei Kanälen, die als Threads des Kanalinitiators ausgeführt werden, wenn der Kanalinitiator gestartet oder erneut gestartet wird und zuerst einen TLS-Kanal ausführt. Wenn der Kanalinitiatorprozess bereits einen TLS-Kanal ausgeführt hat und Sie möchten, dass die Änderung sofort wirksam wird, führen Sie den MQSC-Befehl REFRESH SECURITY TYPE (SSL) aus.
- Für Kanäle, die als Threads eines TCP/IP-Listeners ausgeführt werden, wenn der Listener gestartet oder erneut gestartet wird und zuerst eine Anforderung zum Starten eines TLS-Kanals empfängt. Wenn das Empfangsprogramm bereits einen TLS-Kanal ausgeführt hat und die Änderung sofort wirksam werden soll, führen Sie den MQSC-Befehl REFRESH SECURITY TYPE (SSL) aus.

Verschlüsselungshardware unter IBM i konfigurieren

Gehen Sie wie hier beschrieben vor, um den Cryptographic Coprocessor unter IBM i zu konfigurieren.

Vorbereitende Schritte

Stellen Sie sicher, dass Ihr Benutzerprofil über die Sonderberechtigungen *ALLOBJ und *SECADM verfügt, damit Sie die Coprozessor-Hardware konfigurieren können.

Vorgehensweise

1. Wechseln Sie zu `http://machine.domain:2001` oder `https://machine.domain:2010`, wobei *machine* für den Namen Ihres Computers steht.

   In dem daraufhin aufgerufenen Dialogfenster werden Sie aufgefordert, einen Benutzernamen und ein Kennwort anzugeben.
2. Geben Sie ein gültiges Benutzerprofil und ein gültiges Kennwort für IBM i ein.
3. Rufen Sie Cryptography auf und folgen Sie den entsprechenden Links, um weitere Informationen zu erhalten.

Nächste Schritte

Weitere Informationen zum Konfigurieren des 4767 Cryptographic Coprocessor finden Sie im Abschnitt 4767 Cryptographic Coprocessor .

ALW Mit SSL/TLS unter AIX, Linux, and Windows arbeiten

Auf Systemen mit AIX, Linux, and Windows wird die Unterstützung für Transport Layer Security (TLS) mit IBM MQ installiert.

Weitere Informationen zu Zertifikatsprüfungs-Richtlinien finden Sie im Abschnitt Certificate Validation and Trust Policy Design .

ALW *runmqckm, runmqakmund strmqikm für die Verwaltung digitaler Zertifikate verwenden*

Verwalten Sie Schlüssel und digitale Zertifikate auf Systemen mit AIX, Linux, and Windows über die GUI **strmqikm** (iKeyman) oder über die Befehlszeile mit **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd).

Achtung: Die beiden Befehle **runmqckm** und **strmqikm** basieren auf der IBM MQ Java Runtime Environment (JRE). Wenn die JRE (JRE) nicht installiert ist, erhalten Sie von IBM MQ 9.1 die Nachricht AMQ9183.

- Linux AIX Für **AIX and Linux**-Systeme:
  - Verwenden Sie den Befehl **strmqikm** (iKeyman), um die grafische Benutzerschnittstelle (GUI) von iKeyman zu starten.
  - Mit dem Befehl **runmqckm** können Tasks über die Befehlszeilenschnittstelle ausgeführt werden.
  - Mit dem Befehl **runmqakm** (GSKCapiCmd) können Sie Tasks über die Befehlszeilenschnittstelle 'runmqakm' ausführen. Die Befehlssyntax für **runmqakm** entspricht der Syntax für **runmqckm**.

    Wenn Sie TLS-Zertifikate auf eine FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** anstelle des Befehls **runmqckm** oder **strmqikm** .

  Eine vollständige Beschreibung der Befehlszeilenschnittstellen für die Befehle **runmqckm** und **runmqakm** finden Sie unter Schlüssel und Zertifikate verwalten .

  Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, ist zu beachten, dass es sich bei **runmqckm** und iKeyman um 64-Bit-Programme handelt. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Die 32-Bit-Plattformen Windows und Linux x86 sind die einzigen Ausnahmen, da die Programme iKeyman und **runmqckm** auf diesen Plattformen 32-Bit-Programme sind.

  Weitere Informationen finden Sie unter GSKit: PKCS#11 und IBM MQ JRE-Adressierungsmodus.

Bevor Sie den Befehl **strmqikm** ausführen, um die grafische Benutzerschnittstelle (GUI) von iKeyman zu starten, stellen Sie sicher, dass Sie auf einer Maschine arbeiten, die das X Window System ausführen kann, und dass Sie die folgenden Schritte ausführen:

- Legen Sie die Umgebungsvariable DISPLAY fest. Beispiel:

```
export DISPLAY=mypc:0
```

- Stellen Sie sicher, dass die Umgebungsvariable PATH **/usr/bin** und **/bin** enthält. Dies ist auch für die Befehle **runmqckm** und **runmqakm** erforderlich. Beispiel:

```
export PATH=$PATH:/usr/bin:/bin
```

- Windows Für Systeme mit **Windows** :
  - Verwenden Sie den Befehl **strmqikm** , um die iKeyman -GUI zu starten.
  - Mit dem Befehl **runmqckm** können Tasks über die Befehlszeilenschnittstelle ausgeführt werden.

    Wenn Sie TLS-Zertifikate auf eine FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** anstelle des Befehls **runmqckm** oder **strmqikm** .
  - Verwenden Sie den Befehl **runmqakm -keydb** mit der Option *stashpw* oder *stash*.

    Bei der Verwendung des Befehls **runmqakm -keydb** auf diese Weise (Beispiel:

```
runmqakm -keydb -create -db key.kdb -pw secretpwd -stash
```

    verfügt die sich daraus ergebende Datei `.sth` nicht über Leseberechtigung für die Gruppe `mqm`.

    Die Datei kann nur vom Ersteller gelesen werden. Überprüfen Sie nach dem Erstellen einer Stashdatei mit dem Befehl **runmqakm** die Dateiberechtigung und erteilen Sie dem Service, der den Warteschlangenmanager ausführt, oder einer Gruppe wie beispielsweise der lokalen Gruppe `mqm` die Berechtigung.

ALW Informationen zum Anfordern von TLS-Tracing auf AIX, Linux, and Windows-Systemen finden Sie unter strmqtrc.

Zugehörige Verweise

„runmqckm-und runmqakm-Befehle unter AIX, Linux, and Windows” auf Seite 577
In diesem Abschnitt werden die Befehle **runmqckm** und **runmqakm** anhand der Befehlsobjekte beschrieben.

ALW *Schlüsselrepository unter AIX, Linux, and Windows einrichten*

Sie können ein Schlüsselrepository mithilfe von **strmqikm** einrichten (iKeyman) GUI oder über die Befehlszeile mit Befehlen **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd).

Informationen zu diesem Vorgang

Für eine TLS-Verbindung ist an jedem Ende der Verbindung ein *Schlüsselrepository* erforderlich. Jeder IBM MQ-Warteschlangenmanager und IBM MQ MQI client muss Zugriff auf ein Schlüsselrepository haben. Weitere Informationen finden Sie unter „Das SSL/TLS-Schlüsselrepository” auf Seite 27.

Auf Systemen mit AIX, Linux, and Windows werden digitale Zertifikate in einer Schlüsseldatenbankdatei gespeichert, die über die Benutzerschnittstelle **strmqikm** verwaltet wird, oder mithilfe der Befehle **runmqckm** oder **runmqakm**. Diese digitalen Zertifikate weisen Beschriftungen auf. Eine bestimmte Bezeichnung verknüpft ein persönliches Zertifikat mit einem Warteschlangenmanager oder IBM MQ MQI client. TLS verwendet dieses Zertifikat für Authentifizierungszwecke. Auf Systemen mit AIX, Linux, and Windows verwendet IBM MQ entweder den Wert des Attributs **CERTLABL**, falls dieses festgelegt ist, oder den Standardwert `ibmwebspheremq`, an den der Name des Warteschlangenmanagers oder die Anmelde-ID des IBM MQ MQI client-Benutzers (jeweils in Kleinschreibung) angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

Der Name der Schlüsseldatenbankdatei enthält einen Pfad und einen Stammnamen:

- Auf AIX and Linux-Systemen ist der Standardpfad für einen Queue Manager (wird beim Erstellen eines Queue Managers festgelegt) `/var/mqm/qmgrs/`*`queue_manager_name`*`/ssl`.

  Auf Windows-Systemen lautet der Standardpfad *`MQ_INSTALLATION_PATH`*`\Qmgrs\`*`queue_manager_name`*`\ssl`, wobei *`MQ_INSTALLATION_PATH`* für das Verzeichnis steht, in dem IBM MQ installiert ist. Beispiel: `C:\Programme\IBM\MQ\Qmgrs\QM1\ssl`.

  Der Standardstammname ist `key`. Optional können Sie Ihren eigenen Pfad und Stammnamen auswählen, aber die Erweiterung muss `.kdb` sein.

  Wenn Sie einen eigenen Pfad oder Dateinamen auswählen, legen Sie die Berechtigungen für die Datei fest, um den Zugriff auf diese Datei genau zu steuern.
- Für einen IBM MQ-Client gibt es keinen Standardpfad oder Stammnamen. Der Zugriff auf diese Datei wird direkt gesteuert. Die Erweiterung muss `.kdb` sein.

Erstellen Sie keine Schlüsselrepositorys auf einem Dateisystem, das das Sperren von Dateiebenen nicht unterstützt, z. B. NFS Version 2 auf Linux-Systemen.

Informationen zur Überprüfung und Angabe des Namens der Schlüsseldatenbankdatei finden Sie unter „Position des Schlüsselrepositorys für einen Warteschlangenmanager unter AIX, Linux, and Windows ändern” auf Seite 313. Sie können den Namen der Schlüsseldatenbankdatei entweder vor oder nach der Erstellung der Schlüsseldatenbankdatei angeben.

Die Benutzer-ID, aus der Sie die Befehle **strmqikm** oder **runmqckm**ausführen, muss Schreibzugriff auf das Verzeichnis haben, in dem die Schlüsseldatenbankdatei erstellt oder aktualisiert wird. Für einen Warteschlangenmanager, der das Standardverzeichnis `ssl` verwendet, muss die Benutzer-ID, unter der Sie **strmqikm** oder **runmqckm** ausführen, Mitglied der Gruppe 'mqm' sein. Wenn Sie für eine IBM MQ MQI client-Instanz **strmqikm** oder **runmqckm** mit einer anderen Benutzer-ID als der ausführen, unter der der Client ausgeführt wird, müssen Sie die Dateiberechtigungen ändern, damit IBM MQ MQI client zur Laufzeit auf die Schlüsseldatenbankdatei zugreifen kann. Weitere Informationen finden Sie unter „Unter Windows auf Schlüsseldatenbankdateien zugreifen und diese schützen” auf Seite 311 or „Auf AIX and Linux-Systemen auf Schlüsseldatenbankdateien zugreifen und schützen” auf Seite 311.

In **strmqikm** oder **runmqckm** für GSKit 7.0 werden neue Schlüsseldatenbanken automatisch mit einer Gruppe vordefinierter CA-Zertifikate gefüllt. In **strmqikm** oder **runmqckm** für GSKit 8.0 werden Schlüsseldatenbanken nicht automatisch gefüllt, so dass die Erstkonfiguration sicherer wird, da Sie nur die gewünschten CA-Zertifikate in Ihre Schlüsseldatenbankdatei aufnehmen.

Anmerkung: Da diese Änderung des Verhaltens für GSKit 8.0 dazu führt, dass CA-Zertifikate nicht mehr automatisch dem Repository hinzugefügt werden, müssen Sie Ihre bevorzugten CA-Zertifikate manuell hinzufügen. Diese Verhaltensänderung bietet Ihnen eine differenzierte Kontrolle über die verwendeten CA-Zertifikate. Siehe „Standardzertifikate einer Zertifizierungsstelle unter AIX, Linux, and Windows mit GSKit 8.0 einem leeren Schlüsselrepository hinzufügen” auf Seite 312.

Sie erstellen die Schlüsseldatenbank entweder über die Befehlszeile oder über die Benutzerschnittstelle von **strmqikm** (iKeyman).

Anmerkung: Wenn Sie TLS-Zertifikate auf eine FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** . Die Benutzerschnittstelle **strmqikm** stellt keine FIPS-kompatible Option bereit.

Vorgehensweise

Erstellen Sie über die Befehlszeile eine Schlüsseldatenbank.

1. Führen Sie einen der folgenden Befehle aus:
   - Verwendung von **runmqckm**:

     ```
     runmqckm -keydb -create -db filename -pw password -type cms -stash
     ```

- Verwendung von **runmqakm**:

```
runmqakm -keydb -create -db filename -pw password -type cms
-stash -fips -strong
```

Dabei gilt:

-db *Dateiname*
Gibt den vollständig qualifizierten Dateinamen einer CMS-Schlüsseldatenbank an und muss eine Dateierweiterung von `.kdb` haben.

-pw *password*
Gibt das Kennwort für die CMS-Schlüsseldatenbank an.

-type *cms*
Gibt den Typ der Datenbank an. (Für IBM MQ muss `cms` angegeben werden.)

-stash
Speichert das Kennwort der Schlüsseldatenbank in einer Datei.

-fips
Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 validiert sind. Wenn die ICC-Komponente nicht im FIPS-Modus initialisiert wird, schlägt der Befehl **runmqakm** fehl.

-stark
Überprüft, ob das eingegebene Kennwort die Mindestvoraussetzungen für die Kennwortsicherheit erfüllt. Die Mindestvoraussetzungen für ein Kennwort lauten wie folgt:

- Das Kennwort muss eine Mindestlänge von 14 Zeichen haben.
- Das Kennwort muss mindestens ein Kleinbuchstaben, ein Großbuchstaben und eine Ziffer oder ein Sonderzeichen enthalten. Zu den Sonderzeichen gehören der Stern (*), das Dollarzeichen ($), das Nummernzeichen (#) und das Prozentzeichen (%). Ein Leerzeichen wird als Sonderzeichen klassifiziert.
- Jedes Zeichen kann maximal drei Mal in einem Kennwort vorkommen.
- Es können maximal zwei aufeinanderfolgende Zeichen im Kennwort identisch sein.
- Alle Zeichen sind im Standard-ASCII-Zeichensatz für druckbare Zeichen im Bereich von 0x20 bis 0x7E enthalten.

Alternativ können Sie eine Schlüsseldatenbank über die Benutzerschnittstelle von **strmqikm** (iKeyman) erstellen.

2. Melden Sie sich auf AIX and Linux-Systemen als Rootbenutzer an. Melden Sie sich auf Windows-Systemen als Administrator oder Mitglied der Gruppe MQM an.
3. Starten Sie die Benutzerschnittstelle, indem Sie den Befehl **strmqikm** ausführen.
4. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **New** (Neu).

   Das Fenster Neu wird geöffnet.
5. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
6. Geben Sie in das Feld **Dateiname** einen Dateinamen ein.

   Dieses Feld enthält bereits den Text `key.kdb`. Wenn Ihr Stammname `key` lautet, lassen Sie dieses Feld unverändert. Wenn Sie einen anderen Stammnamen angegeben haben, ersetzen Sie `key` durch Ihren Stammnamen. Sie dürfen die Erweiterung `.kdb` jedoch nicht ändern.
7. Geben Sie in das Feld **Position** den Pfad ein.

   Beispiel:

   - Für einen Queue Manager: `/var/mqm/qmgrs/QM1/ssl` (auf AIX and Linux-Systemen) oder `C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl` (auf Windows-Systemen).

     Der Pfad muss mit dem Wert des **SSLKeyRepository** -Attributs des Warteschlangenmanagers übereinstimmen.

- Für einen IBM MQ-Client: `/var/mqm/ssl` (auf AIX and Linux-Systemen) oder `C:\mqm\ssl` (auf Windows-Systemen).

8. Klicken Sie auf **OK**.

   Das Fenster "Password Prompt" wird geöffnet.

9. Geben Sie ein Kennwort in das Feld **Kennwort** ein, und geben Sie es erneut in das Feld **Kennwort bestätigen** ein.
10. Wählen Sie das Kontrollkästchen **Kennwort in einer Datei speichern** aus.

    Anmerkung: Wenn Sie das Kennwort nicht verstellen, schlagen Versuche zum Starten von TLS-Kanälen fehl, da sie das Kennwort, das für den Zugriff auf die Schlüsseldatenbankdatei erforderlich ist, nicht abrufen können.
11. Klicken Sie auf **OK**.

    Das Fenster Personal Certificates wird geöffnet.
12. Legen Sie die Zugriffsberechtigungen wie unter „Unter Windows auf Schlüsseldatenbankdateien zugreifen und diese schützen" auf Seite 311 oder „Auf AIX and Linux-Systemen auf Schlüsseldatenbankdateien zugreifen und schützen" auf Seite 311 beschrieben fest.

Windows *Unter Windows auf Schlüsseldatenbankdateien zugreifen und diese schützen*

Die Schlüsseldatenbankdateien verfügen möglicherweise nicht über die entsprechenden Zugriffsberechtigungen. Sie müssen den entsprechenden Zugriff auf diese Dateien festlegen.

Legen Sie die Zugriffssteuerung für die Dateien *key*`.kdb`, *key*`.sth`, *key*`.crl`und *key*`.rdb` fest, wobei *Schlüssel* für den Stammnamen Ihrer Schlüsseldatenbank steht, um einer eingeschränkten Gruppe von Benutzern die Berechtigung zu erteilen.

Gehen Sie wie folgt vor, um den Zugriff zu

Vollmacht
BUILTIN\Administrators, NT AUTHORITY\SYSTEM, und der Benutzer, der die Datenbankdateien erstellt hat.

Leseberechtigung
Nur für einen WS-Manager die lokale Gruppe mqm. Dabei wird davon ausgegangen, dass der MCA unter einer Benutzer-ID in der Gruppe mqm ausgeführt wird.

Für einen Client die Benutzer-ID, unter der der Clientprozess ausgeführt wird.

Linux AIX *Auf AIX and Linux-Systemen auf Schlüsseldatenbankdateien zugreifen und schützen*

Die Schlüsseldatenbankdateien verfügen möglicherweise nicht über die entsprechenden Zugriffsberechtigungen. Sie müssen den entsprechenden Zugriff auf diese Dateien festlegen.

Für einen Warteschlangenmanager legen Sie die Berechtigungen für die Schlüsseldatenbankdateien fest, damit Warteschlangenmanager und Kanalprozesse sie lesen können, wenn dies erforderlich ist, andere Benutzer können sie jedoch nicht lesen oder ändern. Normalerweise benötigt der mqm-Benutzer Leseberechtigungen. Wenn Sie die Schlüsseldatenbankdatei erstellt haben, indem Sie sich als mqm-Benutzer anmelden, sind die Berechtigungen wahrscheinlich ausreichend; wenn Sie nicht der mqm-Benutzer, sondern ein anderer Benutzer in der Gruppe mqm waren, müssen Sie wahrscheinlich anderen Benutzern in der Gruppe mqm Leseberechtigungen erteilen.

In ähnlicher Weise legen Sie für einen Client die Berechtigungen für die Schlüsseldatenbankdateien fest, damit Clientanwendungsprozesse sie lesen können, wenn dies erforderlich ist, andere Benutzer können sie jedoch nicht lesen oder ändern. Normalerweise benötigt der Benutzer, unter dem der Clientprozess ausgeführt wird, Leseberechtigungen. Wenn Sie die Schlüsseldatenbankdatei erstellt haben, indem Sie sich als dieser Benutzer anmelden, sind die Berechtigungen wahrscheinlich ausreichend. Wenn Sie nicht der Client-Prozessbenutzer waren, sondern ein anderer Benutzer in dieser Gruppe, müssen Sie wahrscheinlich anderen Benutzern in der Gruppe Leseberechtigungen erteilen.

Legen Sie die Berechtigungen für die Dateien *key*.kdb, *key*.sth, *key*.crlund *key*.rdb fest. Dabei steht *Schlüssel* für den Stammnamen Ihrer Schlüsseldatenbank, für Lesen und Schreiben für den Dateieigner und für Lesen für die Gruppe 'mqm' oder für die Clientbenutzergruppe (-rw-r -----).

ALW *Standardzertifikate einer Zertifizierungsstelle unter AIX, Linux, and Windows mit GSKit 8.0 einem leeren Schlüsselrepository hinzufügen*

Führen Sie diese Prozedur aus, um eine oder mehrere der Standardzertifikat einer Zertifizierungsstelle zu einem leeren Schlüsselrepository mit GSKit 8.0hinzuzufügen.

In GSKit 7.0wurde das Verhalten beim Erstellen eines neuen Schlüsselrepositorys automatisch in einer Gruppe von Standard-CA-Zertifikaten für die allgemein verwendeten Zertifizierungsstellen hinzugefügt. Für GSKit 8.0 hat sich dieses Verhalten geändert, so dass CA-Zertifikate nicht mehr automatisch dem Repository hinzugefügt werden. Der Benutzer muss jetzt CA-Zertifikate manuell in das Schlüsselrepository hinzufügen.

strmqikm verwenden

Führen Sie die folgenden Schritte auf der Maschine aus, auf der Sie das CA-Zertifikat hinzufügen möchten:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm** (unter AIX, Linux, and Windows).
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, der Sie das Zertifikat hinzufügen möchten, z. B. key.kdb.
6. Klicken Sie auf **Öffnen** . Das Fenster "Password Prompt" wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** . Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Wählen Sie im Feld **Key database content** die Option **Signer Certificates** aus.
9. Klicken Sie auf **Populate** . Das Fenster Zertifizierungsstelle hinzufügen wird geöffnet.
10. Die CA-Zertifikate, die dem Repository hinzugefügt werden können, werden in einer hierarchischen Baumstruktur angezeigt. Wählen Sie den Eintrag der höchsten Ebene für die Organisation aus, deren CA-Zertifikate Sie vertrauen möchten, um die vollständige Liste der gültigen CA-Zertifikate anzuzeigen.
11. Wählen Sie in der Liste die CA-Zertifikate aus, denen Sie vertrauen möchten, und klicken Sie auf **OK** . Die Zertifikate werden dem Schlüsselrepository hinzugefügt.

Verwenden der Befehlszeile

Verwenden Sie die folgenden Befehle, um eine Liste hinzuzufügen, und fügen Sie anschließend CA-Zertifikate mithilfe von **runmqckm** hinzu:

- Geben Sie den folgenden Befehl aus, um die Standardzertifikate der Zertifizierungsstellen zusammen mit den Organisationen aufzulisten, die sie ausgeben:

```
runmqckm -cert -listsigners
```

- Geben Sie den folgenden Befehl aus, um alle CA-Zertifikate für die Organisation hinzuzufügen, die im Feld *label* angegeben ist:

```
runmqckm -cert -populate -db filename -pw password -label label
```

Dabei gilt:

-db *filename*	ist der vollständig qualifizierte Pfadname der Schlüsseldatenbank.
-pw *password*	ist das Kennwort für die Schlüsseldatenbank.
-label *label*	Ist der Kennsatz, der dem Zertifikat zugeordnet ist.

Anmerkung: Das Hinzufügen eines CA-Zertifikats zu einem Schlüsselrepository führt dazu, dass IBM MQ alle persönlichen Zertifikate, die von diesem CA-Zertifikat signiert wurden, als vertrauenswürdig betrachtet. Berücksichtigen Sie sorgfältig die Zertifizierungsinstanzen, denen Sie vertrauen möchten, und fügen Sie nur die Gruppe von CA-Zertifikaten hinzu, die für die Authentifizierung Ihrer Clients und Manager erforderlich sind. Es wird nicht empfohlen, die vollständige Gruppe von Standardzertifikaten von CA-Zertifikaten hinzuzufügen, es sei denn, dies ist eine definitive Voraussetzung für Ihre Sicherheitsrichtlinie.

ALW *Schlüsselrepository für einen Warteschlangenmanager unter AIX, Linux, and Windows ermitteln*

Verwenden Sie diese Prozedur, um die Position der Schlüsseldatenbankdatei Ihres WS-Managers abzurufen.

Vorgehensweise

1. Zeigen Sie die Attribute des WS-Managers mit einem der folgenden MQSC-Befehle an:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

   Sie können die Attribute Ihres Warteschlangenmanagers auch mit dem IBM MQ Explorer oder den PCF-Befehlen anzeigen.
2. Untersuchen Sie die Befehlsausgabe für den Pfad und den Stammnamen der Schlüsseldatenbankdatei.
   Zum Beispiel:
   a. Unter AIX and Linux: `/var/mqm/qmgrs/QM1/ssl/key`. Dabei steht `/var/mqm/qmgrs/QM1/ssl` für den Pfad und `key` für den Stammnamen.
   b. Unter Windows: *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl\key`. Dabei steht *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl` für den Pfad und `key` für den Stammnamen. *MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

ALW *Position des Schlüsselrepositorys für einen Warteschlangenmanager unter AIX, Linux, and Windows ändern*

Sie können die Position der Schlüsseldatenbankdatei Ihres WS-Managers mit verschiedenen Mitteln ändern, einschließlich des MQSC-Befehls ALTER QMGR.

Sie können die Position der Schlüsseldatenbankdatei Ihres WS-Managers ändern, indem Sie den WebSphere MQ-Scriptbefehl ALTER QMGR verwenden, um das Schlüsselrepository-Attribut des WS-Managers festzulegen. Beispiel unter AIX and Linux:

```
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/QM1/ssl/MyKey')
```

Der vollständig qualifizierte Dateiname der Schlüsseldatenbankdatei lautet `/var/mqm/qmgrs/QM1/ssl/MyKey.kdb`

Unter Windows:

```
ALTER QMGR SSLKEYR('C:\Programme\IBM\MQ\Qmgrs\QM1\ssl\Mykey')
```

Der vollständig qualifizierte Dateiname der Schlüsseldatenbankdatei lautet `C:\Programme\IBM\MQ\Qmgrs\QM1\ssl\Mykey.kdb`

Achtung: Stellen Sie sicher, dass Sie die Erweiterung `.kdb` nicht in den Dateinamen im Schlüsselwort SSLKEYR aufnehmen, da der WS-Manager diese Erweiterung automatisch anhängt.

Sie können auch die Attribute des Warteschlangenmanagers mit dem IBM MQ Explorer oder mit PCF-Befehlen ändern.

Wenn Sie den Speicherort für die Schlüsseldatenbankdatei eines WS-Managers ändern, werden die Zertifikate nicht automatisch an den neuen Speicherort übertragen. Wenn die Schlüsseldatenbankdatei, auf die Sie jetzt zugreifen, eine neue Schlüsseldatenbankdatei ist, müssen Sie sie mit den von Ihnen benötigten CA-und persönlichen Zertifikaten füllen, wie in „Persönliches Zertifikat unter AIX, Linux, and Windows in ein Schlüsselrepository importieren” auf Seite 329 beschrieben.

ALW *Schlüsselrepository für einen IBM MQ MQI client unter AIX, Linux, and Windows suchen*

Die Position des Schlüsselrepositorys wird durch die Variable MQSSLKEYR angegeben oder im MQCONNX-Aufruf angegeben.

Prüfen Sie die Umgebungsvariable MQSSLKEYR, um die Position der Schlüsseldatenbankdatei für Ihren IBM MQ MQI client zu finden. Beispiel:

```
echo $MQSSLKEYR
```

Sie sollten auch Ihre Anwendung überprüfen, da der Name der Schlüsseldatenbankdatei auch in einem MQCONNX-Aufruf gesetzt werden kann, wie unter „Position des Schlüsselrepositorys für einen IBM MQ MQI client unter AIX, Linux, and Windows angeben” auf Seite 314 beschrieben. Der in einem MQCONNX-Aufruf festgelegte Wert überschreibt den Wert von MQSSLKEYR.

ALW *Position des Schlüsselrepositorys für einen IBM MQ MQI client unter AIX, Linux, and Windows angeben*

Für einen IBM MQ MQI client gibt es kein standardmäßiges Schlüsselrepository. Sie können die Position auf eine der beiden Arten angeben. Stellen Sie sicher, dass auf die Schlüsseldatenbankdatei nur von bestimmten Benutzern oder Administratoren zugegriffen werden kann, um ein unbefugtes Kopieren auf andere Systeme zu verhindern.

Sie können die Position der Schlüsseldatenbankdatei für Ihren IBM MQ MQI client auf zwei Arten angeben:

- Definieren Sie die Umgebungsvariable MQSSLKEYR. Beispiel unter AIX and Linux:

  ```
  export MQSSLKEYR=/var/mqm/ssl/key
  ```

  Die Schlüsseldatenbankdatei hat den vollständig qualifizierten Dateinamen:

  ```
  /var/mqm/ssl/key.kdb
  ```

  Unter Windows:

  ```
  set MQSSLKEYR=C:\Programme\IBM\MQ\ssl\key
  ```

  Die Schlüsseldatenbankdatei hat den vollständig qualifizierten Dateinamen:

  ```
  C:\Programme\IBM\MQ\ssl\key.kdb
  ```

  Anmerkung: Die Erweiterung `.kdb` ist ein obligatorischer Teil des Dateinamens, wird aber nicht als Teil des Werts der Umgebungsvariablen angegeben.

- Geben Sie den Pfad und den Stammnamen der Schlüsseldatenbankdatei im Feld *KeyRepository* der MQSCO-Struktur an, wenn eine Anwendung einen MQCONNX-Aufruf vornimmt. Weitere Informationen zur Verwendung der MQSCO-Struktur in MQCONNX finden Sie unter Übersicht für MQSCO .

ALW *Zeitpunkt, an dem Änderungen an Zertifikaten oder dem Zertifikatsspeicher unter AIX, Linux, and Windows wirksam werden*

Wenn Sie die Zertifikate in einem Zertifikatsspeicher oder in der Position des Zertifikatsspeichers ändern, werden die Änderungen in Abhängigkeit vom Typ des Kanals und der Ausführung des Kanals wirksam.

Änderungen an den Zertifikaten in der Schlüsseldatenbankdatei und an dem Schlüsselrepository-Attribut werden in den folgenden Situationen wirksam:

- Wenn ein neuer abgehender Einzelkanalprozess zuerst einen TLS-Kanal ausführt.
- Wenn ein neuer eingehender TCP/IP-Einzelkanalprozess zuerst eine Anforderung zum Starten eines TLS-Kanals empfängt.
- Wenn der MQSC-Befehl REFRESH SECURITY TYPE (SSL) ausgegeben wird, um die TLS-Umgebung zu aktualisieren.
- Bei Clientanwendungsprozessen, wenn die letzte TLS-Verbindung in dem Prozess geschlossen wird. Die nächste TLS-Verbindung wird die Zertifikatänderungen übernehmen.
- Für Kanäle, die als Threads in einem Prozess-Pooling-Prozess (amqrmppa) ausgeführt werden, wenn der Prozess-Pooling-Prozess gestartet oder erneut gestartet wird und zuerst einen TLS-Kanal ausführt. Wenn der Prozess-Pooling-Prozess bereits einen TLS-Kanal ausgeführt hat und Sie möchten, dass die Änderung sofort wirksam wird, führen Sie den MQSC-Befehl REFRESH SECURITY TYPE (SSL) aus.
- Bei Kanälen, die als Threads des Kanalinitiators ausgeführt werden, wenn der Kanalinitiator gestartet oder erneut gestartet wird und zuerst einen TLS-Kanal ausführt. Wenn der Kanalinitiatorprozess bereits einen TLS-Kanal ausgeführt hat und Sie möchten, dass die Änderung sofort wirksam wird, führen Sie den MQSC-Befehl REFRESH SECURITY TYPE (SSL) aus.
- Für Kanäle, die als Threads eines TCP/IP-Listeners ausgeführt werden, wenn der Listener gestartet oder erneut gestartet wird und zuerst eine Anforderung zum Starten eines TLS-Kanals empfängt. Wenn das Empfangsprogramm bereits einen TLS-Kanal ausgeführt hat und die Änderung sofort wirksam werden soll, führen Sie den MQSC-Befehl REFRESH SECURITY TYPE (SSL) aus.

Sie können die IBM MQ-TLS-Umgebung auch mit dem IBM MQ Explorer oder mit PCF-Befehlen aktualisieren.

ALW *Selbst signiertes persönliches Zertifikat unter AIX, Linux, and Windows erstellen*

Sie können ein selbst signiertes Zertifikat mit **strmqikm** (iKeyman) erstellen. GUI oder über die Befehlszeile mit **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd).

Anmerkung: IBM MQ unterstützt keine SHA-3- oder SHA-5-Algorithmen. Sie können die Namen der digitalen Signaturalgorithmen SHA384WithRSA und SHA512WithRSA verwenden, da beide Algorithmen zu Mitgliedern der SHA-2-Familie gehören.

Die Namen der digitalen Signaturalgorithmen SHA3WithRSA und SHA5WithRSA werden nicht weiter unterstützt, da sie eine abgekürzte Form von SHA384WithRSA bzw. SHA512WithRSA sind.

Weitere Informationen darüber, warum Sie selbst signierte Zertifikate verwenden möchten, finden Sie im Abschnitt Selbst signierte Zertifikate für die gegenseitige Authentifizierung von zwei Warteschlangenmanagern verwenden .

Nicht alle digitalen Zertifikate können mit allen CipherSpecs verwendet werden. Stellen Sie sicher, dass Sie ein Zertifikat erstellen, das mit den CipherSpecs kompatibel ist, die Sie verwenden müssen. IBM MQ unterstützt drei verschieden Typen von CipherSpec. Weitere Informationen finden Sie unter „Interoperabilität von Elliptic Curve und RSA CipherSpecs” auf Seite 50 im Abschnitt „Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ” auf Seite 49

Um die CipherSpecs des Typs 1 (die mit Namen, die mit ECDHE_ECDSA_beginnen) zu verwenden, müssen Sie den Befehl **runmqakm** verwenden, um das Zertifikat zu erstellen, und Sie müssen einen Elliptic Curve ECDSA-Signaturalgorithmusparameter angeben, z. B. **-sig_alg** EC_ecdsa_with_SHA384.

Eine Liste der mit dem **-sig_alg** -Hashalgorithmus verfügbaren Optionen finden Sie unter „runmqckm- und runmqakm-Optionen unter AIX, Linux, and Windows" auf Seite 591 .

Informationen bei Verwendung der

- Grafischen Benutzerschnittstelle finden Sie unter „Benutzerschnittstelle strmqikm verwenden" auf Seite 316
- Befehlszeile finden Sie unter „Verwenden der Befehlszeile" auf Seite 317

ALW *Benutzerschnittstelle **strmqikm** verwenden*

Sie können ein persönliches Zertifikat mithilfe der Benutzerschnittstelle **strmqikm** (iKeyman) erstellen.

Informationen zu diesem Vorgang

strmqikm stellt keine FIPS-kompatible Option bereit. Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm** .

Vorgehensweise

Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat für Ihren Warteschlangenmanager oder den IBM MQ MQI client mithilfe der grafische Benutzerschnittstelle zu erstellen:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm**.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen).

   Das Fenster **Open** (Öffnen) wird angezeigt.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, aus der die Anforderung generiert werden soll, z. B. key.kdb.
6. Klicken Sie auf **OK**.

   Das Fenster **Password Prompt** wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** .

   Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Klicken Sie im Menü **Create** auf **New Self-Signed Certificate** (Neues selbst signiertes Zertifikat). Das Fenster "Neues selbst signiertes Zertifikat erstellen" wird angezeigt.
9. Geben Sie in das Feld **Schlüsselkennsatz** die Zertifikatsbezeichnung ein.

   Bei der Bezeichnung handelt es sich um den Wert des Attributs **CERTLABL**, falls dieses festgelegt ist, oder um den Standardwert ibmwebspheremq, an den der Name des Warteschlangenmanagers oder die Anmeldebenutzer-ID des IBM MQ MQI clients (jeweils in Kleinschreibung) angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .
10. Geben Sie im Feld **Definierter Name** einen Wert für ein beliebtes Feld ein oder wählen Sie einen Wert in den Feldern **Name des alternativen Namens** aus.
11. Geben Sie für die übrigen Felder entweder die Standardwerte an, oder geben Sie neue Werte ein oder wählen Sie neue Werte aus.

    Weitere Informationen zu definierten Namen finden Sie unter „Definierte Namen" auf Seite 14.
12. Klicken Sie auf **OK**.

    In der Liste **Persönliche Zertifikate** wird die Bezeichnung des selbst signierten persönlichen Zertifikats angezeigt, das Sie erstellt haben.

ALW *Verwenden der Befehlszeile*

Sie können ein persönliches Zertifikat über die Befehlszeile mit den Befehlen **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd) erstellen. Wenn Sie SSL-oder TLS-Zertifikate auf FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** .

Vorgehensweise

Erstellen Sie ein selbst signiertes persönliches Zertifikat mit dem Befehl **runmqckm** oder **runmqakm** (GSKCapiCmd).

- Verwendung von **runmqckm**:

```
runmqckm -cert -create -db filename -pw password -label label
         -dn distinguished_name -size key_size
         -x509version version -expire days -sig_alg algorithm
```

  Anstelle von -dn *distinguished_name*können Sie -san_dnsname *DNS_names*, -san_emailaddr *email_addresses*oder -san_ipaddr *IP_addresses*verwenden.

- Verwendung von **runmqakm**:

```
runmqakm -cert -create -db filename -pw password -label label
         -dn distinguished_name -size key_size
         -x509version version -expire days -fips -sig_alg algorithm
```

Dabei gilt:

-db *Dateiname*
Gibt den vollständig qualifizierten Dateinamen einer CMS-Schlüsseldatenbank an.

-pw *password*
Gibt das Kennwort für die CMS-Schlüsseldatenbank an.

-label *Bezeichnung*
Gibt den Schlüsselkennsatz an, der dem Zertifikat zugeordnet ist. Die Bezeichnung ist entweder der Wert des Attributs **CERTLABL** , wenn es festgelegt ist, oder der Standardwert ibmwebspheremq mit dem Namen des Warteschlangenmanagers oder der angehängten IBM MQ MQI client Benutzer-ID, alles in Kleinbuchstaben. Weitere Informationen finden Sie in „Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen” auf Seite 29 .

-dn *definierter_Name*
Gibt den definierten X.500-Namen in doppelte Anführungszeichen an. Mindestens ein Attribut ist erforderlich. Sie können mehrere OU-und DC-Attribute angeben.

Anmerkung: Die Tools **runmqckm** und **runmqakm** beziehen sich auf das Postleitzahlenattribut POSTALCODE und nicht auf PC. Geben Sie immer POSTALCODE im Parameter **-dn** an, wenn Sie diese Zertifikatsmanagementbefehle verwenden, um Zertifikate mit einer Postleitzahl anzufordern.

-size *Schlüsselgröße*
Gibt die Schlüsselgröße an. Wenn Sie **runmqckm**verwenden, kann der Wert 512 oder 1024lauten. Wenn Sie **runmqakm**verwenden, kann der Wert 512, 1024oder 2048sein.

x509version *Version*
Die Version des zu erstellenden X.509-Zertifikats. Der Wert kann 1, 2 oder 3 sein. Der Standardwert ist 3.

-file *Dateiname*
Gibt den Dateinamen für die Zertifikatsanforderung an.

-expire *Tage*
Die Verfallszeit in Tagen des Zertifikats. Der Standardwert ist 365 Tage für ein Zertifikat.

-fips
Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Es wird nur die FIPS-ICC-Komponente verwendet, und diese Komponente muss im FIPS-Modus erfolgreich initialisiert werden. Im FIPS-Modus

verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente nicht im FIPS-Modus initialisiert wird, schlägt der Befehl **runmqakm** fehl.

-sig_alg
Gibt für **runmqckm**den asymmetrischen Signaturalgorithmus an, der für die Erstellung des Schlüsselpaars des Eintrags verwendet wird. Folgende Werte sind möglich: MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256_WITH_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA, SHAWithRSA. Der Standardwert ist SHA1WithRSA .

-sig_alg
Gibt für **runmqakm**den Hashalgorithmus an, der beim Erstellen einer Zertifikatsanforderung verwendet wird. Dieser Hashing-Algorithmus wird verwendet, um die Signatur zu erstellen, die der neu erstellten Zertifikatsanforderung zugeordnet ist. Der Wert kann md5, MD5_WITH_RSA, MD5WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, sha1, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, sha224, SHA224_WITH_RSA, SHA224WithDSA, SHA224WithECDSA, SHA224WithRSA, sha256, SHA256_WITH_RSA, SHA256WithDSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithRSA, sha384, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, sha512, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHAWithDSA, SHAWithRSA, EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224, EC_ecdsa_with_SHA256, EC_ecdsa_with_SHA384, oder EC_ecdsa_with_SHA512 sein. Der Standardwert ist SHA1WithRSA .

-san_dnsname *DNS_names*
Gibt eine durch Kommas oder Leerzeichen getrennte Liste mit DNS-Namen für den Eintrag an, der erstellt wird.

-san_emailaddr *email_addresses*
Gibt eine durch Kommas begrenzte oder durch Leerzeichen getrennte Liste von E-Mail-Adressen für den zu erstellenden Eintrag an.

-san_ipaddr *IP_addresses*
Gibt eine durch Kommas begrenzte oder durch Leerzeichen getrennte Liste mit IP-Adressen für den zu erstellenden Eintrag an.

ALW *Persönliches Zertifikat unter AIX, Linux, and Windows anfordern*

Sie können ein persönliches Zertifikat über die **strmqikm** anfordern (iKeyman) GUI oder über die Befehlszeile mit den Befehlen **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd). Wenn Sie SSL-oder TLS-Zertifikate auf FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** .

Informationen zu diesem Vorgang

Sie können ein persönliches Zertifikat über die grafische Benutzerschnittstelle **strmqikm** oder über die Befehlszeile anfordern, wobei Sie Folgendes berücksichtigen müssen:

- IBM MQ unterstützt keine SHA-3- oder SHA-5-Algorithmen. Sie können die Namen der digitalen Signaturalgorithmen SHA384WithRSA und SHA512WithRSA verwenden, da beide Algorithmen zu Mitgliedern der SHA-2-Familie gehören.
- Die Namen der digitalen Signaturalgorithmen SHA3WithRSA und SHA5WithRSA werden nicht weiter unterstützt, da sie eine abgekürzte Form von SHA384WithRSA bzw. SHA512WithRSA sind.
- Nicht alle digitalen Zertifikate können mit allen CipherSpecs verwendet werden. Stellen Sie sicher, dass Sie ein Zertifikat anfordern, das mit den CipherSpecs kompatibel ist, die Sie verwenden müssen. IBM MQ unterstützt drei verschieden Typen von CipherSpec. Weitere Informationen finden Sie unter „Interoperabilität von Elliptic Curve und RSA CipherSpecs” auf Seite 50 im Abschnitt „Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ” auf Seite 49
- Wenn Sie die CipherSpecs des Typs 1 (mit Namen, die mit ECDHE_ECDSA_beginnen) verwenden möchten, müssen Sie den Befehl **runmqakm** verwenden, um das Zertifikat anzufordern, und Sie müssen einen Elliptic Curve ECDSA-Signaturalgorithmusparameter angeben, z. B. **-sig_alg** EC_ecdsa_with_SHA384.

Eine Liste der mit dem **-sig_alg** -Hashalgorithmus verfügbaren Optionen finden Sie unter „runmqckm-und runmqakm-Optionen unter AIX, Linux, and Windows” auf Seite 591 .

- Nur der Befehl **runmqakm** stellt eine FIPS-konforme Option bereit.
- Wenn Sie Verschlüsselungshardware verwenden, lesen Sie den Abschnitt „Anfordern eines persönlichen Zertifikats für Ihre PKCS #11-Hardware” auf Seite 339.

Informationen bei Verwendung der

- Grafischen Benutzerschnittstelle finden Sie unter „Benutzerschnittstelle strmqikm verwenden” auf Seite 319
- Befehlszeile finden Sie unter „Verwenden der Befehlszeile” auf Seite 320

ALW *Benutzerschnittstelle **strmqikm** verwenden*

Sie können ein persönliches Zertifikat über die GUI von **strmqikm** (iKeyman) anfordern. Wenn Sie SSL- oder TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm** .

Informationen zu diesem Vorgang

strmqikm stellt keine FIPS-kompatible Option bereit. Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm** .

Vorgehensweise

Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat zu beantragen, indem Sie die iKeyman-Benutzerschnittstelle verwenden:

1. Starten Sie die Benutzerschnittstelle, indem Sie den Befehl **strmqikm** verwenden.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen).

   Das Fenster **Öffnen** wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, aus der die Anforderung generiert werden soll, z. B. `key.kdb`.
6. Klicken Sie auf **Öffnen** .

   Das Fenster **Password Prompt** wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** .

   Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Klicken Sie im Menü **Erstellen** auf **Neue Zertifikatsanforderung** . Das Fenster **Neuen Schlüssel und Zertifikatsanforderung erstellen** wird geöffnet.
9. Geben Sie in das Feld **Schlüsselkennsatz** die Zertifikatsbezeichnung ein.

   Bei der Bezeichnung handelt es sich um den Wert des Attributs **CERTLABL**, falls dieses festgelegt ist, oder um den Standardwert `ibmwebspheremq`, an den der Name des Warteschlangenmanagers oder die Anmeldebenutzer-ID des IBM MQ MQI clients (jeweils in Kleinschreibung) angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .
10. Geben Sie im Feld **Definierter Name** einen Wert für ein beliebtes Feld ein oder wählen Sie einen Wert in den Feldern **Name des alternativen Namens** aus. Übernehmen Sie für die übrigen Felder entweder die Standardwerte, oder geben Sie neue Werte ein oder wählen Sie sie aus.

    Weitere Informationen zu definierten Namen finden Sie unter „Definierte Namen” auf Seite 14.
11. Geben Sie im Feld **Geben Sie den Namen einer Datei ein, in die das Zertifikatsanforderung gespeichert werden soll** entweder den Standardwert `certreq.arm` ein, oder geben Sie einen neuen Wert mit einem vollständigen Pfad ein.
12. Klicken Sie auf **OK**.

Ein Bestätigungsfenster wird angezeigt.

13. Klicken Sie auf **OK**.

    In der Liste **Persönliche Zertifikatsanforderungen** wird die Bezeichnung der neuen persönlichen Zertifikatsanforderung angezeigt, die Sie erstellt haben. Die Zertifikatsanforderung wird in der unter Punkt „11" auf Seite 319 ausgewählten Datei gespeichert.

14. Fordern Sie das neue persönliche Zertifikat an, indem Sie die Datei an eine Zertifizierungsstelle (CA) senden oder indem Sie die Datei in das Anforderungsformular auf der Website für die CA kopieren.

ALW *Verwenden der Befehlszeile*

Sie können ein persönliches Zertifikat über die Befehlszeile mit den Befehlen **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd) anfordern. Wenn Sie SSL-oder TLS-Zertifikate auf FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** .

Vorgehensweise

Fordern Sie ein selbst signiertes persönliches Zertifikat mit dem Befehl **runmqckm** oder **runmqakm** (GSKCapiCmd) an.

- Verwendung von **runmqckm**:

```
runmqckm -certreq -create -db filename -pw
password -label label
         -dn distinguished_name -size key_size
        -file filename -sig_alg algorithm
```

  Anstelle von `-dn distinguished_name`können Sie `-san_dsname DNS_names`, `-san_emailaddr email_addresses`oder `-san_ipaddr IP_addresses`verwenden.

- Verwendung von **runmqakm**:

```
runmqakm -certreq -create -db filename -pw
password -label label
         -dn distinguished_name -size key_size
         -file filename -fips -sig_alg algorithm
```

Dabei gilt:

-db *Dateiname*
Gibt den vollständig qualifizierten Dateinamen einer CMS-Schlüsseldatenbank an.

-pw *password*
Gibt das Kennwort für die CMS-Schlüsseldatenbank an.

-label *Bezeichnung*
Gibt den Schlüsselkennsatz an, der dem Zertifikat zugeordnet ist. Die Bezeichnung ist entweder der Wert des Attributs **CERTLABL** , wenn es festgelegt ist, oder der Standardwert `ibmwebspheremq` mit dem Namen des Warteschlangenmanagers oder der angehängten IBM MQ MQI client Benutzer-ID, alles in Kleinbuchstaben. Weitere Informationen finden Sie in „Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen" auf Seite 29 .

-dn *definierter_Name*
Gibt den definierten X.500-Namen in doppelte Anführungszeichen an. Mindestens ein Attribut ist erforderlich. Sie können mehrere OU-und DC-Attribute angeben.

Anmerkung: Die Tools **runmqckm** und **runmqakm** beziehen sich auf das Postleitzahlenattribut POSTALCODE und nicht auf PC. Geben Sie immer POSTALCODE im Parameter **-dn** an, wenn Sie diese Zertifikatsmanagementbefehle verwenden, um Zertifikate mit einer Postleitzahl anzufordern.

-size *Schlüsselgröße*
Gibt die Schlüsselgröße an. Wenn Sie **runmqckm**verwenden, kann der Wert 512 oder 1024lauten. Wenn Sie **runmqakm**verwenden, kann der Wert 512, 1024oder 2048sein.

-file *Dateiname*
Gibt den Dateinamen für die Zertifikatsanforderung an.

-fips
Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 validiert sind. Wenn die ICC-Komponente nicht im FIPS-Modus initialisiert wird, schlägt der Befehl **runmqakm** fehl.

-sig_alg
Gibt für **runmqckm**den asymmetrischen Signaturalgorithmus an, der für die Erstellung des Schlüsselpaars des Eintrags verwendet wird. Folgende Werte sind möglich: MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256_WITH_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA, SHAWithRSA. Der Standardwert ist SHA1WithRSA .

-sig_alg
Gibt für **runmqakm**den Hashalgorithmus an, der beim Erstellen einer Zertifikatsanforderung verwendet wird. Dieser Hashing-Algorithmus wird verwendet, um die Signatur zu erstellen, die der neu erstellten Zertifikatsanforderung zugeordnet ist. Der Wert kann md5, MD5_WITH_RSA, MD5WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, sha1, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, sha224, SHA224_WITH_RSA, SHA224WithDSA, SHA224WithECDSA, SHA224WithRSA, sha256, SHA256_WITH_RSA, SHA256WithDSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithRSA, sha384, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, sha512, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHAWithDSA, SHAWithRSA, EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224, EC_ecdsa_with_SHA256, EC_ecdsa_with_SHA384, oder EC_ecdsa_with_SHA512 sein. Der Standardwert ist SHA1WithRSA .

-san_dnsname *DNS_names*
Gibt eine durch Kommas oder Leerzeichen getrennte Liste mit DNS-Namen für den Eintrag an, der erstellt wird.

-san_emailaddr *email_addresses*
Gibt eine durch Kommas begrenzte oder durch Leerzeichen getrennte Liste von E-Mail-Adressen für den zu erstellenden Eintrag an.

-san_ipaddr *IP_addresses*
Gibt eine durch Kommas begrenzte oder durch Leerzeichen getrennte Liste mit IP-Adressen für den zu erstellenden Eintrag an.

Nächste Schritte

Übergeben Sie eine Zertifikatsanforderung an eine CA. Weitere Informationen finden Sie im Abschnitt „Persönliche Zertifikate in einem Schlüsselrepository unter AIX, Linux, and Windows empfangen” auf Seite 323.

ALW *Vorhandenes persönliches Zertifikat unter AIX, Linux, and Windows verlängern*

Sie können ein persönliches Zertifikat über die **strmqikm** verlängern (iKeyman) GUI oder über die Befehlszeile mit den Befehlen **runmqckm** (iKeycmd) oder **runmqakm** (GSKCapiCmd).

Informationen zu diesem Vorgang

Wenn Sie eine größere Schlüsselgröße für Ihre persönlichen Zertifikate verwenden müssen, können Sie ein vorhandenes Zertifikat nicht verlängern. Sie müssen Ihren vorhandenen Schlüssel ersetzen, indem Sie die in „Persönliches Zertifikat unter AIX, Linux, and Windows anfordern” auf Seite 318 beschriebenen Schritte ausführen, um eine neue Zertifikatsanforderung zu erstellen, die die von Ihnen benötigten Schlüsselgrößen verwendet.

Ein persönliches Zertifikat weist ein Ablaufdatum auf, nach dessen Ablauf das Zertifikat nicht mehr verwendet werden kann. In dieser Übung wird beschrieben, wie ein vorhandenes persönliches Zertifikat vor dem Ablauf erneuert wird.

*Benutzerschnittstelle **strmqikm** verwenden*

Informationen zu diesem Vorgang

strmqikm stellt keine FIPS-kompatible Option bereit. Wenn Sie TLS-Zertifikate auf eine FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** .

Vorgehensweise

Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat zu beantragen, indem Sie die Benutzerschnittstelle **strmqikm** verwenden:

1. Starten Sie die Benutzerschnittstelle mit dem Befehl **strmqikm** unter AIX, Linux, and Windows.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen).

   Das Fenster **Öffnen** wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, aus der die Anforderung generiert werden soll, z. B. `key.kdb`.
6. Klicken Sie auf **Öffnen** .

   Das Fenster **Password Prompt** wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** .

   Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Wählen Sie im Dropdown-Auswahlmenü **Persönliche Zertifikate** aus, und wählen Sie das Zertifikat aus der Liste aus, das Sie erneuern möchten.
9. Klicken Sie auf **Anforderung erneut erstellen** . Schaltfläche geklickt haben.

   Es wird ein Fenster geöffnet, in dem Sie den Dateinamen und die Informationen zur Dateiposition eingeben können.
10. Geben Sie im Feld **Dateiname** entweder den Standardwert `certreq.arm` an oder geben Sie einen neuen Wert ein, einschließlich des vollständigen Dateipfads.
11. Klicken Sie auf **OK**. Die Zertifikatsanforderung wird in der Datei gespeichert, die Sie in Schritt „9” auf Seite 322 ausgewählt haben.
12. Fordern Sie das neue persönliche Zertifikat an, indem Sie die Datei an eine Zertifizierungsstelle (CA) senden oder indem Sie die Datei in das Anforderungsformular auf der Website für die CA kopieren.

Verwenden der Befehlszeile

Vorgehensweise

Verwenden Sie die folgenden Befehle, um ein persönliches Zertifikat anzufordern, indem Sie entweder den Befehl **runmqckm** oder **runmqakm** verwenden:

- Verwendung von **runmqckm**:

```
runmqckm -certreq -recreate -db filename -pw
password -label label
-target filename
```

- runmqakm wird verwendet:

```
runmqakm -certreq -recreate -db filename -pw
password -label label
-target filename
```

Dabei gilt:

-db *Dateiname*
Gibt den vollständig qualifizierten Dateinamen einer CMS-Schlüsseldatenbank an.

-pw *password*
Gibt das Kennwort für die CMS-Schlüsseldatenbank an.

-target *Dateiname*
Gibt den Dateinamen für die Zertifikatsanforderung an.

Nächste Schritte

Nachdem Sie das signierte persönliche Zertifikat von der Zertifizierungsstelle erhalten haben, können Sie es mithilfe der in „Persönliche Zertifikate in einem Schlüsselrepository unter AIX, Linux, and Windows empfangen” auf Seite 323 beschriebenen Schritte zu Ihrer Schlüsseldatenbank hinzufügen.

ALW *Persönliche Zertifikate in einem Schlüsselrepository unter AIX, Linux, and Windows empfangen*

Verwenden Sie diese Prozedur, um ein persönliches Zertifikat in der Schlüsseldatenbankdatei zu empfangen. Das Schlüsselrepository muss mit dem Repository identisch sein, in dem Sie die Zertifikatsanforderung erstellt haben.

Nachdem die CA Ihnen ein neues persönliches Zertifikat gesendet hat, fügen Sie es der Schlüsseldatenbankdatei hinzu, aus der Sie die neue Zertifikatsanforderung generiert haben. Wenn die Zertifizierungsstelle das Zertifikat als Teil einer E-Mail-Nachricht sendet, kopieren Sie das Zertifikat in eine separate Datei.

`strmqikm` verwenden

Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm**. **strmqikm** stellt keine FIPS-kompatible Option bereit.

Stellen Sie sicher, dass die zu importierende Zertifikatsdatei über Schreibzugriff für den aktuellen Benutzer verfügt. Führen Sie anschließend die folgenden Schritte für einen Warteschlangenmanager oder einen IBM MQ MQI client aus, um ein persönliches Zertifikat in der Schlüsseldatei zu empfangen:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm**.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, der das Zertifikat hinzugefügt werden soll, z. B. key.kdb.
6. Klicken Sie auf **Öffnen** , und klicken Sie dann auf **OK** . Das Fenster "Password Prompt" wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** . Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt. Wählen Sie die Ansicht **Personal Certificates** aus.
8. Klicken Sie auf **Empfangen** . Das Fenster 'Receive Certificate from a File' (Zertifikat aus einer Datei empfangen) wird angezeigt.
9. Geben Sie den Namen und die Position der Zertifikatsdatei für das neue persönliche Zertifikat ein, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.

10. Klicken Sie auf **OK**. Wenn Sie bereits über ein persönliches Zertifikat in Ihrer Schlüsseldatenbank verfügen, wird ein Fenster geöffnet, in dem Sie gefragt werden, ob Sie den Schlüssel festlegen möchten, den Sie als Standardschlüssel in der Datenbank hinzufügen möchten.
11. Klicken Sie auf **Ja** oder **Nein**. Das Fenster "Enter a Label" wird geöffnet.
12. Klicken Sie auf **OK**. Im Feld **Persönliche Zertifikate** wird die Bezeichnung des neuen persönlichen Zertifikats angezeigt, das Sie hinzugefügt haben.

Verwenden der Befehlszeile

Verwenden Sie einen der folgenden Befehle, um einer Schlüsseldatenbankdatei ein persönliches Zertifikat hinzuzufügen:

- Mit **runmqckm** :

```
runmqckm -cert -receive -file filename -db filename -pw password
         -format ascii
```

- Mit **runmqakm** :

```
runmqakm -cert -receive -file filename -db filename -pw password -fips
```

Dabei gilt:

-file *Dateiname*
Gibt den vollständig qualifizierten Dateinamen des persönlichen Zertifikats an.

-db *Dateiname*
Gibt den vollständig qualifizierten Dateinamen einer CMS-Schlüsseldatenbank an.

-pw *password*
Gibt das Kennwort für die CMS-Schlüsseldatenbank an.

-format *ascii*
Gibt das Format des Zertifikats an. Der Wert kann `ascii` für Base64-encoded ASCII oder `binary` für binäre DER-Daten sein. Der Standardwert ist `ascii`.

-fips
Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente im FIPS-Modus nicht initialisiert wird, schlägt der Befehl **runmqakm** fehl.

Wenn Sie Verschlüsselungshardware verwenden, lesen Sie den Abschnitt „Persönliches Zertifikat in Ihrer PKCS #11-Hardware empfangen” auf Seite 340.

ALW *Zertifikat einer Zertifizierungsstelle aus einem Schlüsselrepository unter AIX, Linux, and Windows extrahieren*

Gehen Sie wie folgt vor, um ein CA-Zertifikat zu extrahieren.

`strmqikm` verwenden

Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm**. **strmqikm** (iKeyman) stellt keine FIPS-kompatible Option bereit.

Führen Sie die folgenden Schritte auf dem System aus, aus dem Sie das CA-Zertifikat extrahieren möchten:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm**.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.

4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, aus der Sie extrahieren möchten, z. B. `key.kdb`.
6. Klicken Sie auf **Öffnen** . Das Fenster "Password Prompt" wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** . Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Wählen Sie im Feld **Key database content** die Option **Signer Certificates** (Unterzeichnerzertifikate) aus und wählen Sie das Zertifikat aus, das Sie extrahieren möchten.
9. Klicken Sie auf **Extrahieren** . Das Fenster 'Zertifikat in eine Datei extrahieren' wird geöffnet.
10. Wählen Sie den **Datentyp** des Zertifikats aus, z. B. **Base64-codierte ASCII-Daten** für eine Datei mit der Erweiterung " `.arm` ".
11. Geben Sie den Namen und die Position der Zertifikatsdatei ein, in der das Zertifikat gespeichert werden soll, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.
12. Klicken Sie auf **OK**. Das Zertifikat wird in die von Ihnen angegebene Datei geschrieben.

Verwenden der Befehlszeile

Verwenden Sie die folgenden Befehle, um ein CA-Zertifikate mit den Befehlen **runmqckm** oder **runmqakm** zu extrahieren:

```
runmqckm -cert -extract -db filename -pw password -label label
         -target filename -format ascii
```

oder

```
runmqakm -cert -extract -db filename -pw password -label label
         -target filename -format ascii -fips
```

Dabei gilt:

`-db` *filename*	ist der vollständig qualifizierte Pfadname einer CMS-Schlüsseldatenbank.
`-pw` *password*	ist das Kennwort für die CMS-Schlüsseldatenbank.
`-label` *label*	Ist der Kennsatz, der dem Zertifikat zugeordnet ist.
`-target` *filename*	ist der Name der Zieldatei.
`-format` *ascii*	ist das Format des Zertifikats. Der Wert kann `ascii` für Base64-encoded ASCII oder `binary` für binäre DER-Daten sein. Der Standardwert ist `ascii`.
`-fips`	Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente nicht im FIPS-Modus initialisiert wird, schlägt der Befehl **runmqakm** fehl.

ALW *Öffentlichen Teil eines selbst signierten Zertifikats unter AIX, Linux, and Windows aus einem Schlüsselrepository extrahieren*

Führen Sie die folgende Prozedur aus, um den öffentlichen Teil eines selbst signierten Zertifikats zu extrahieren.

`strmqikm` verwenden

Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm**. **strmqikm** (iKeyman) stellt keine FIPS-kompatible Option bereit.

Führen Sie die folgenden Schritte auf dem System aus, von dem aus Sie den öffentlichen Teil eines selbst signierten Zertifikats extrahieren möchten:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm**.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, aus der das Zertifikat extrahiert werden soll, z. B. `key.kdb`.
6. Klicken Sie auf **OK**. Das Fenster "Password Prompt" wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** . Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Wählen Sie im Feld **Key database content** die Option **Personal Certificates** (Persönliche Zertifikate) aus und wählen Sie das Zertifikat aus.
9. Klicken Sie auf **Extract Certificate** (Zertifikat extrahieren Das Fenster 'Zertifikat in eine Datei extrahieren' wird geöffnet.
10. Wählen Sie den **Datentyp** des Zertifikats aus, z. B. **Base64-codierte ASCII-Daten** für eine Datei mit der Erweiterung " `.arm` ".
11. Geben Sie den Namen und die Position der Zertifikatsdatei ein, in der das Zertifikat gespeichert werden soll, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.
12. Klicken Sie auf **OK**. Das Zertifikat wird in die von Ihnen angegebene Datei geschrieben. Beachten Sie, dass beim Extrahieren (und nicht Exportieren) eines Zertifikats nur der öffentliche Teil des Zertifikats enthalten ist, so dass ein Kennwort nicht erforderlich ist.

Verwenden der Befehlszeile

Verwenden Sie die folgenden Befehle, um den öffentlichen Teil eines selbst signierten Zertifikats mit **runmqckm** oder **runmqakm** zu extrahieren:

- Mit „runmqckm“:

```
runmqckm -cert -extract -db filename -pw password -label label -target filename
         -format ascii
```

- runmqakm wird verwendet:

```
runmqakm -cert -extract -db filename -pw password -label label
            -target filename -format ascii -fips
```

Dabei gilt:

`-db` *filename*	ist der vollständig qualifizierte Pfadname einer CMS-Schlüsseldatenbank.
`-pw` *password*	ist das Kennwort für die CMS-Schlüsseldatenbank.
`-label` *label*	Ist der Kennsatz, der dem Zertifikat zugeordnet ist.
`-target` *filename*	ist der Name der Zieldatei.
`-format` *ascii*	ist das Format des Zertifikats. Der Wert kann `ascii` für Base64-encoded ASCII oder `binary` für binäre DER-Daten sein. Der Standardwert ist `ascii`.

`-fips`	Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente im FIPS-Modus nicht initialisiert wird, schlägt der Befehl **runmqakm** fehl.

ALW *CA-Zertifikat oder öffentlichen Teil eines selbst signierten Zertifikats unter AIX, Linux, and Windows einem Schlüsselrepository hinzufügen*

Gehen Sie wie folgt vor, um ein CA-Zertifikat oder den öffentlichen Teil eines selbst signierten Zertifikats zum Schlüsselrepository hinzuzufügen.

Wenn sich das Zertifikat, das Sie hinzufügen möchten, in einer Zertifikatskette befindet, müssen Sie auch alle Zertifikate hinzufügen, die sich in der Kette darüber befinden. Sie müssen die Zertifikate in strikt absteigender Reihenfolge beginnend mit dem Stammverzeichnis, gefolgt von dem CA-Zertifikat, das unmittelbar unter der Kette in der Kette liegt, und so weiter hinzufügen.

Wenn die folgenden Anweisungen auf ein CA-Zertifikat verweisen, gelten sie auch für den öffentlichen Teil eines selbst signierten Zertifikats.

Anmerkung: Sie müssen sicherstellen, dass das Zertifikat in ASCII (UTF-8) oder Binärformat (DER) codiert ist, da IBM Global Secure Toolkit (GSKit) keine Zertifikate mit anderen Codierungsarten unterstützt.

strmqikm verwenden

Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm**. **strmqikm** stellt keine FIPS-kompatible Option bereit.

Führen Sie die folgenden Schritte auf der Maschine aus, auf der Sie das CA-Zertifikat hinzufügen möchten:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm**.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, der das Zertifikat hinzugefügt werden soll, z. B. `key.kdb`.
6. Klicken Sie auf **OK**. Das Fenster "Password Prompt" wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** . Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Wählen Sie im Feld **Key database content** die Option **Signer Certificates** aus.
9. Klicken Sie auf **Hinzufügen**. Das Fenster CA-Zertifikat aus einem Datei hinzufügen wird geöffnet.
10. Geben Sie den Namen und die Position der Zertifikatsdatei ein, in der das Zertifikat gespeichert ist, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.
11. Klicken Sie auf **OK**. Das Fenster "Enter a Label" wird geöffnet.
12. Geben Sie im Fenster "Enter a Label" den Namen des Zertifikats ein.
13. Klicken Sie auf **OK**. Das Zertifikat wird der Schlüsseldatenbank hinzugefügt.

Verwenden der Befehlszeile

Verwenden Sie einen der folgenden Befehle, um einer Schlüsseldatenbankdatei ein Zertifikat einer Zertifizierungsstelle hinzuzufügen:

- Mit **runmqckm** :

```
runmqckm -cert -add -db filename -pw password -label label
         -file filename -format ascii
```

- Mit **runmqakm** :

```
runmqakm -cert -add -db filename -pw password -label label
         -file filename -format ascii -fips
```

Dabei gilt:

-db *Dateiname*
Gibt den vollständig qualifizierten Dateinamen der CMS-Schlüsseldatenbank an.

-pw *password*
Gibt das Kennwort für die CMS-Schlüsseldatenbank an.

-label *Bezeichnung*
Gibt die Bezeichnung an, die dem Zertifikat zugeordnet ist.

-file *Dateiname*
Gibt den Dateinamen der Datei an, die das Zertifikat enthält.

-format *ascii*
Gibt das Format des Zertifikats an. Der Wert kann `ascii` für Base64-encoded ASCII oder `binary` für binäre DER-Daten sein. Der Standardwert ist `ascii`.

-fips
Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente im FIPS-Modus nicht initialisiert wird, schlägt der Befehl **runmqakm** fehl.

ALW *Persönliches Zertifikat aus einem Schlüsselrepository unter AIX, Linux, and Windows exportieren*

Gehen Sie wie folgt vor, um ein persönliches Zertifikat zu exportieren.

strmqikm verwenden

Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm**. **strmqikm** (iKeyman) stellt keine FIPS-kompatible Option bereit.

Führen Sie die folgenden Schritte auf dem System aus, von dem aus Sie das persönliche Zertifikat exportieren möchten:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm**.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, aus der das Zertifikat exportiert werden soll, z. B. `key.kdb`.
6. Klicken Sie auf **Öffnen** . Das Fenster "Password Prompt" wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** . Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Wählen Sie im Feld **Key database content** die Option **Personal Certificates** (Persönliche Zertifikate) aus und wählen Sie das Zertifikat aus, das Sie exportieren möchten.
9. Klicken Sie auf **Exportieren/Importieren** . Das Fenster "Schlüssel exportieren/importieren" wird geöffnet.
10. Wählen Sie **Schlüssel exportieren** aus.
11. Wählen Sie den **Schlüsseldatentyp** des Zertifikats aus, das exportiert werden soll, z. B. **PKCS12** .

12. Geben Sie den Dateinamen und die Position ein, an die das Zertifikat exportiert werden soll, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.
13. Klicken Sie auf **OK**. Das Fenster "Password Prompt" wird geöffnet. Beachten Sie, dass beim Exportieren (und nicht Extrahieren) eines Zertifikats sowohl der öffentliche als auch der private Teil des Zertifikats enthalten sind. Aus diesem Grund ist die exportierte Datei durch ein Kennwort geschützt. Wenn Sie ein Zertifikat extrahieren, ist nur der öffentliche Teil des Zertifikats enthalten, so dass ein Kennwort nicht erforderlich ist.
14. Geben Sie ein Kennwort in das Feld **Kennwort** ein, und geben Sie es erneut in das Feld **Kennwort bestätigen** ein.
15. Klicken Sie auf **OK**. Das Zertifikat wird in die von Ihnen angegebene Datei exportiert.

Verwenden der Befehlszeile

Exportieren Sie ein persönliches Zertifikat mit dem Befehl **runmqckm** oder mit dem Befehl **runmqakm**:

```
runmqckm -cert -export -db filename -pw password -label label -type cms
        -target filename -target_pw password -target_type pkcs12
```

oder

```
runmqakm -cert -export -db filename -pw password -label label -type cms
        -target filename -target_pw password -target_type pkcs12
        -encryption strong | weak -fips
```

Dabei gilt:

-db *filename*	ist der vollständig qualifizierte Pfadname der CMS-Schlüsseldatenbank.
-encryption	ist die Stärke der Verschlüsselung, die im Befehl zum Exportieren von Zertifikaten verwendet wird. Der Wert kann stark oder schwach sein. Der Standardwert ist strong .
-fips	Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente nicht im FIPS-Modus initialisiert wird, schlägt der Befehl **runmqakm** fehl.
-pw *password*	ist das Kennwort für die CMS-Schlüsseldatenbank.
-label *label*	Ist der Kennsatz, der dem Zertifikat zugeordnet ist.
-type *cms*	ist der Typ der Datenbank.
-target *filename*	ist der vollständig qualifizierte Pfadname der Zieldatei.
-target_pw *password*	ist das Kennwort zum Verschlüsseln des Zertifikats.
-target_type *pkcs12*	ist der Typ des Zertifikats.

ALW *Persönliches Zertifikat unter AIX, Linux, and Windows in ein Schlüsselrepository importieren*

Gehen Sie wie folgt vor, um ein persönliches Zertifikat zu importieren:

Vor dem Importieren eines persönlichen Zertifikats im PKCS-#12-Format in die Schlüsseldatenbankdatei muss zunächst die vollständige CA-Ausstellerzertifikatskette zur Schlüsseldatenbankdatei hinzugefügt werden (siehe „CA-Zertifikat oder öffentlichen Teil eines selbst signierten Zertifikats unter AIX, Linux, and Windows einem Schlüsselrepository hinzufügen” auf Seite 327).

PKCS#12-Dateien sollten als temporär betrachtet und nach der Verwendung gelöscht werden.

strmqikm verwenden

Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm** . **strmqikm** stellt keine FIPS-kompatible Option bereit.

Führen Sie die folgenden Schritte auf der Maschine aus, auf die Sie das persönliche Zertifikat importieren möchten:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm**.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird angezeigt.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, der das Zertifikat hinzugefügt werden soll, z. B. key.kdb.
6. Klicken Sie auf **Öffnen** . Das Fenster "Password Prompt" wird angezeigt.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** . Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Wählen Sie im Feld **Key database content** die Option **Personal Certificates** (Persönliche Zertifikate) aus.
9. Wenn Zertifikate in der Ansicht "Persönliche Zertifikate" vorhanden sind, führen Sie die folgenden Schritte aus:
   a. Klicken Sie auf **Exportieren/Importieren** . Das Fenster "Schlüssel exportieren/importieren" wird angezeigt.
   b. Wählen Sie **Schlüssel importieren** aus.
10. Wenn keine Zertifikate in der Ansicht "Persönliche Zertifikate" vorhanden sind, klicken Sie auf **Importieren** .
11. Wählen Sie den **Schlüsseldatentyp** des Zertifikats aus, das Sie importieren möchten, z. B. PKCS12.
12. Geben Sie den Namen und die Position der Zertifikatsdatei ein, in der das Zertifikat gespeichert ist, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.
13. Klicken Sie auf **OK**. Das Fenster "Password Prompt" wird angezeigt.
14. Geben Sie in das Feld **Kennwort** das Kennwort ein, das beim Exportieren des Zertifikats verwendet wurde.
15. Klicken Sie auf **OK**. Das Fenster "Beschriftungen ändern" wird angezeigt. Sie können die Beschriftungen von Zertifikaten ändern, die importiert werden, wenn z. B. bereits ein Zertifikat mit derselben Bezeichnung in der Zielschlüsseldatenbank vorhanden ist. Das Ändern der Zertifikatsbezeichnungen hat keine Auswirkungen auf die Validierung der Zertifikatskette. Für die Zuordnung des Zertifikats zu einem bestimmten Warteschlangenmanager oder einem IBM MQ MQI client verwendet IBM MQ den Wert des Attributs **CERTLABL**, falls dieses festgelegt ist, oder den Standardwert ibmwebspheremq, an den der Name des Warteschlangenmanagers oder die Anmelde-ID des IBM MQ MQI client-Benutzers (jeweils in Kleinschreibung) angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .
16. Wenn Sie eine Bezeichnung ändern möchten, wählen Sie die gewünschte Bezeichnung in der Liste **Eine zu änderliche Bezeichnung auswählen** aus. Die Bezeichnung wird in das Eingabefeld **Geben Sie ein neues Kennsatz eingeben** kopiert. Ersetzen Sie den Beschriftungstext durch die neue Bezeichnung, und klicken Sie auf **Anwenden** .
17. Der Text im Eingabefeld **Neuen Kennsatz eingeben** wird wieder in das Feld **Zu änderndem Kennsatz auswählen** kopiert, wobei der ursprünglich ausgewählte Kennsatz ersetzt wird und das entsprechende Zertifikat so neu angeordnet wird.
18. Wenn Sie alle Beschriftungen geändert haben, die geändert werden mussten, klicken Sie auf **OK** . Das Fenster 'Change Labels' wird geschlossen und das ursprüngliche IBM Key Management-Fenster mit

den Feldern **Personal Certificates** und **Signer Certificates** wird wieder mit den korrekt gekennzeichneten Zertifikaten angezeigt.

19. Das Zertifikat wird in die Zielschlüsseldatenbank importiert.

Verwenden der Befehlszeile

Für den Import eines persönlichen Zertifikats mit **runmqckm** verwenden Sie den folgenden Befehl:

```
runmqckm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label
```

Wenn Sie ein persönliches Zertifikat mit **runmqakm** importieren möchten, verwenden Sie den folgenden Befehl:

```
runmqakm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label -fips
```

Dabei gilt:

`-file` *filename*	ist der vollständig qualifizierte Dateiname der Datei, die das PKCS#12-Zertifikat enthält.
`-pw` *password*	ist das Kennwort für das PKCS#12-Zertifikat.
`-type` *pkcs12*	ist der Typ der Datei.
`-target` *filename*	ist der Name der Ziel-CMS-Schlüsseldatenbank.
`-target_pw` *password*	ist das Kennwort für die CMS-Schlüsseldatenbank.
`-target_type` *cms*	Der Typ der Datenbank, der durch `-target` angegeben wird.
`-label` *label*	ist die Bezeichnung des Zertifikats, das aus der Quellenschlüsseldatenbank importiert werden soll.
`-new_label` *label*	ist der Kennsatz, der dem Zertifikat in der Zieldatenbank zugeordnet wird. Wenn Sie die Option `-new_label` nicht angeben, wird standardmäßig dieselbe Option wie die Option `-label` verwendet.
`-fips`	Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente im FIPS-Modus nicht initialisiert wird, schlägt der Befehl **runmqakm** fehl.

In **runmqckm** gibt es keinen Befehl, mit dem Zertifikatsbezeichnungen direkt geändert werden können. Führen Sie die folgenden Schritte aus, um eine Zertifikatsbezeichnung zu ändern:

1. Exportieren Sie das Zertifikat mit dem Befehl **-cert -export** in eine PKCS#12-Datei. Geben Sie die vorhandene Zertifikatsbezeichnung für die Option `-label` an.
2. Entfernen Sie die vorhandene Kopie des Zertifikats aus der ursprünglichen Schlüsseldatenbank mit dem Befehl **-cert -delete** .
3. Importieren Sie das Zertifikat mit dem Befehl **-cert -import** aus der PKCS#12-Datei. Geben Sie die alte Bezeichnung für die Option `-label` und die erforderliche neue Bezeichnung für die Option `-new_label` an. Das Zertifikat wird mit der erforderlichen Bezeichnung zurück in die Schlüsseldatenbank importiert.

ALW *Persönliches Zertifikat aus einer Microsoft.pfx-Datei importieren*

Führen Sie die folgenden Schritte aus, um eine Microsoft.pfx-Datei unter AIX, Linux, and Windows zu importieren.

Eine .pfx-Datei kann zwei Zertifikate enthalten, die sich auf denselben Schlüssel beziehen. Ein Zertifikat ist ein persönliches Zertifikat oder ein Site-Zertifikat (mit einem öffentlichen und einem privaten Schlüssel). Das andere ist ein CA-Zertifikat (Unterzeichnerzertifikat), das nur einen öffentlichen Schlüssel enthält. Diese Zertifikate können nicht in derselben CMS-Schlüsseldatenbankdatei koexistieren, sodass nur eine von ihnen importiert werden kann. Außerdem wird der "aussagekräftiger Name" oder die Bezeichnung nur an das Unterzeichnerzertifikat angehängt.

Das persönliche Zertifikat wird durch eine vom System generierte eindeutige Benutzer-ID (Unique User Identifier-UUID) identifiziert. In diesem Abschnitt wird der Import eines persönlichen Zertifikats aus einer PFX-Datei beim Kennzeichnen dieses Abschnitts mit dem Namen angezeigt, der zuvor dem CA-Zertifikat (Unterzeichnerzertifikat) zugeordnet wurde. Die ausstellenden CA-Zertifikate (Unterzeichnerzertifikate) sollten bereits zur Zielschlüsseldatenbank hinzugefügt werden. Beachten Sie, dass PKCS#12-Dateien als temporär betrachtet und nach der Verwendung gelöscht werden sollten.

Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat aus einer Quellenpfx-Schlüsseldatenbank zu importieren:

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **`strmqikm`**. Das Fenster 'IBM Key Management' wird angezeigt.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird angezeigt.
3. Wählen Sie einen Schlüsseldatenbanktyp **PKCS12** aus.
4. **Es wird empfohlen, vor der Ausführung dieses Schritts eine Sicherung der PFX-Datenbank zu erstellen.** Wählen Sie die pfx-Schlüsseldatenbank aus, die Sie importieren wollen. Klicken Sie auf **Öffnen** . Das Fenster "Password Prompt" wird angezeigt.
5. Geben Sie das Kennwort für die Schlüsseldatenbank ein und klicken Sie auf **OK** . Das Fenster 'IBM Key Management' wird angezeigt. In der Titelleiste wird der Name der ausgewählten PFX-Schlüsseldatenbankdatei angezeigt, die angibt, dass die Datei geöffnet und bereit ist.
6. Wählen Sie in der Liste **Unterzeichnerzertifikate** aus. Der "Anzeigename" des erforderlichen Zertifikats wird in der Anzeige "Signer Certificates" als Bezeichnung angezeigt.
7. Wählen Sie den Kennsatzeintrag aus und klicken Sie auf **Löschen** , um das Unterzeichnerzertifikat zu entfernen. Das Fenster Bestätigen wird angezeigt.
8. Klicken Sie auf **Ja** . Die ausgewählte Bezeichnung wird nicht mehr in der Anzeige "Signer Certificates" angezeigt.
9. Wiederholen Sie die Schritte 6, 7 und 8 für alle Unterzeichnerzertifikate.
10. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird angezeigt.
11. Wählen Sie die CMS-Datenbank des Zielschlüssels aus, in die die PFX-Datei importiert wird. Klicken Sie auf **Öffnen** . Das Fenster "Password Prompt" wird angezeigt.
12. Geben Sie das Kennwort für die Schlüsseldatenbank ein und klicken Sie auf **OK** . Das Fenster 'IBM Key Management' wird angezeigt. In der Titelleiste wird der Name der ausgewählten Schlüsseldatenbankdatei angezeigt, die angibt, dass die Datei geöffnet und bereit ist.
13. Wählen Sie in der Liste **Persönliche Zertifikate** aus.
14. Wenn Zertifikate in der Ansicht "Persönliche Zertifikate" vorhanden sind, führen Sie die folgenden Schritte aus:
    a. Klicken Sie auf **Schlüssel exportieren/importieren** . Das Fenster "Schlüssel exportieren/importieren" wird angezeigt.
    b. Wählen Sie **Import** from Choose Action Type (Aktionstyp auswählen)
15. Wenn keine Zertifikate in der Ansicht "Persönliche Zertifikate" vorhanden sind, klicken Sie auf **Importieren** .
16. Wählen Sie die PKCS12-Datei aus.
17. Geben Sie den Namen der pfx-Datei ein, die in Schritt 4 verwendet wird. Klicken Sie auf **OK**. Das Fenster "Password Prompt" wird angezeigt.

18. Geben Sie das gleiche Kennwort an, das Sie beim Löschen des Unterzeichnerzertifikats angegeben haben. Klicken Sie auf **OK**.
19. Das Fenster "Beschriftungen ändern" wird angezeigt (da es nur ein einziges Zertifikat für den Import verfügbar sein sollte). Die Bezeichnung des Zertifikats muss eine UUID sein, die ein Format `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` hat.
20. Wenn Sie die Bezeichnung ändern möchten, wählen Sie die UUID in der Anzeige **Select a label to change:** aus. Die Bezeichnung wird in das Feld **Geben Sie ein neues Kennsatz eingeben:** repliziert. Ersetzen Sie den Beschriftungstext durch den Namen des in Schritt 7 gelöschten aussagekräftigen Namens, und klicken Sie auf **Anwenden**. Der aussagekräftige Name muss entweder der Wert des Attributs IBM MQ **CERTLABL** sein, wenn dieser gesetzt ist, oder der Standardwert `ibmwebspheremq` mit dem Namen des angehängten Queue Managers oder der angehängten IBM MQ MQI client-Benutzer-Anmelde-ID in Kleinschreibung sein. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .
21. Klicken Sie auf **OK**. Das Fenster 'Change Labels' (Bezeichnungen ändern) ist jetzt entfernt und das ursprüngliche IBM Key Management-Fenster wird wieder angezeigt, in dem die Anzeigen 'Personal Certificates' (Persönliche Zertifikate) und 'Signer Certificates' (Unterzeichnerzertifikate) durch die ordnungsgemäß gekennzeichneten persönlichen Zertifikate aktualisiert wurden.
22. Das persönliche PFX-Zertifikat wird nun in die (Ziel-) Datenbank importiert.

Es ist nicht möglich, eine Zertifikatsbezeichnung mit **runmqckm** oder **runmqakm** zu ändern.

Verwenden der Befehlszeile

Für den Import eines persönlichen Zertifikats mit **runmqckm** verwenden Sie den folgenden Befehl:

```
runmqckm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label -pfx
```

Wenn Sie ein persönliches Zertifikat mit **runmqakm** importieren möchten, verwenden Sie den folgenden Befehl:

```
runmqakm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label -fips -pfx
```

Dabei gilt:

`-file` *filename*	ist der vollständig qualifizierte Dateiname der Datei, die das PKCS#12-Zertifikat enthält.
`-pw` *password*	ist das Kennwort für das PKCS#12-Zertifikat.
`-type` *pkcs12*	ist der Typ der Datei.
`-target` *filename*	ist der Name der Ziel-CMS-Schlüsseldatenbank.
`-target_pw` *password*	ist das Kennwort für die CMS-Schlüsseldatenbank.
`-target_type` *cms*	Der Typ der Datenbank, der durch `-target` angegeben wird.
`-label` *label*	ist die Bezeichnung des Zertifikats, das aus der Quellenschlüsseldatenbank importiert werden soll.
`-new_label` *label*	ist der Kennsatz, der dem Zertifikat in der Zieldatenbank zugeordnet wird. Wenn Sie die Option `-new_label` nicht angeben, wird standardmäßig dieselbe Option wie die Option `-label` verwendet.
`-fips`	Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente im FIPS-Modus nicht initialisiert wird, schlägt der Befehl **runmqakm** fehl.

`-pfx`	Gibt das PFX-Dateiformat an.

In **runmqckm** gibt es keinen Befehl, mit dem Zertifikatsbezeichnungen direkt geändert werden können. Führen Sie die folgenden Schritte aus, um eine Zertifikatsbezeichnung zu ändern:

1. Exportieren Sie das Zertifikat mit dem Befehl **-cert -export** in eine PKCS#12-Datei. Geben Sie die vorhandene Zertifikatsbezeichnung für die Option `-label` an.
2. Entfernen Sie die vorhandene Kopie des Zertifikats aus der ursprünglichen Schlüsseldatenbank mit dem Befehl **-cert -delete** .
3. Importieren Sie das Zertifikat mit dem Befehl **-cert -import** aus der PKCS#12-Datei. Geben Sie die alte Bezeichnung für die Option `-label` und die erforderliche neue Bezeichnung für die Option `-new_label` an. Das Zertifikat wird mit der erforderlichen Bezeichnung zurück in die Schlüsseldatenbank importiert.

ALW *Persönlichem Zertifikat aus einer PKCS#7-Datei importieren*

Die Tools **strmqikm** (iKeyman) und **runmqckm** (iKeycmd) unterstützen keine PKCS #7-Dateien (`.p7b`). Mit dem Tool **runmqakm** können Sie Zertifikate aus einer PKCS #7 -Datei unter AIX, Linux, and Windowsimportieren.

Verwenden Sie den folgenden Befehl, um ein CA-Zertifikat aus einer PKCS #7-Datei hinzuzufügen:

```
runmqakm -cert -add -db filename -pw password -type cms -file filename
-label label
```

`-db` *filename*	steht für den vollständig qualifizierten Dateinamen der CMS-Schlüsseldatenbank.
`-pw` *password*	ist das Kennwort für die Schlüsseldatenbank.
`-type` *cms*	ist der Typ der Schlüsseldatenbank.
`-file` *filename*	ist der Name der PKCS#7-Datei.
`-label` *label*	ist der Kennsatz, dem das Zertifikat in der Zieldatenbank zugeordnet ist. Das erste Zertifikat verwendet die angegebene Bezeichnung. Alle anderen Zertifikate, falls vorhanden, sind mit ihrem Betreffnamen gekennzeichnet.

Verwenden Sie den folgenden Befehl, um ein persönliches Zertifikat aus einer PKCS #7-Datei zu importieren:

```
runmqakm -cert -import -db filename -pw password -type pkcs7 -target filename
-target_pw password -target_type cms -label label -new_label label
```

`-db` *filename*	ist der vollständig qualifizierte Dateiname der Datei, die das PKCS #7-Zertifikat enthält.
`-pw` *password*	ist das Kennwort für das PKCS #7-Zertifikat.
`-type` *pkcs7*	ist der Typ der Datei.
`-target` *filename*	ist der Name der Zielschlüsseldatenbank.
`-target_pw` *password*	ist das Kennwort für die Zielschlüsseldatenbank.
`-target_type` *cms*	Der Typ der Datenbank, der durch `-target` angegeben wird.
`-label` *label*	ist die Bezeichnung des Zertifikats, das importiert werden soll.
`-new_label` *label*	ist der Kennsatz, der dem Zertifikat in der Zieldatenbank zugeordnet wird. Wenn Sie die Option `-new_label` nicht angeben, wird standardmäßig dieselbe Option wie die Option `-label` verwendet.

ALW *Zertifikat unter AIX, Linux, and Windows aus einem Schlüsselrepository löschen*

Verwenden Sie diese Prozedur, um persönliche Zertifikate oder CA-Zertifikate zu entfernen.

`strmqikm` verwenden

Wenn Sie TLS-Zertifikate in einer Weise verwalten müssen, die FIPS-konform ist, verwenden Sie den Befehl **runmqakm**. **strmqikm** (iKeyman) stellt keine FIPS-kompatible Option bereit.

1. Starten Sie die grafische Benutzerschnittstelle mit dem Befehl **strmqikm**.
2. Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen). Das Fenster Öffnen wird geöffnet.
3. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **CMS** (Certificate Management System) aus.
4. Klicken Sie auf **Durchsuchen** , um zu dem Verzeichnis zu navigieren, das die Schlüsseldatenbankdateien enthält.
5. Wählen Sie die Schlüsseldatenbankdatei aus, aus der das Zertifikat gelöscht werden soll, z. B. `key.kdb`.
6. Klicken Sie auf **Öffnen** . Das Fenster "Password Prompt" wird geöffnet.
7. Geben Sie das Kennwort ein, das Sie bei der Erstellung der Schlüsseldatenbank festgelegt haben, und klicken Sie auf **OK** . Der Name Ihrer Schlüsseldatenbankdatei wird im Feld **Dateiname** angezeigt.
8. Wählen Sie in der Dropdown-Liste die Option **Personal Certificates** (Persönliche Zertifikate) oder **Signer Certificates** (Unterzeichnerzertifikate) aus.
9. Wählen Sie das Zertifikat aus, das Sie löschen möchten.
10. Wenn noch keine Kopie des Zertifikats vorhanden ist und es gespeichert werden soll, klicken Sie auf **Exportieren/Importieren**, und exportieren Sie es (siehe „Persönliches Zertifikat aus einem Schlüsselrepository unter AIX, Linux, and Windows exportieren” auf Seite 328).
11. Klicken Sie bei ausgewähltes Zertifikat auf **Löschen** . Das Fenster "Bestätigen" wird geöffnet
12. Klicken Sie auf **Ja** . Im Feld **Personal Certificates** wird die Bezeichnung des gelöschten Zertifikats nicht mehr angezeigt.

Verwenden der Befehlszeile

Verwenden Sie die folgenden Befehle, um ein Zertifikat mit dem Befehl **runmqckm** oder mit dem Befehl **runmqakm** zu löschen:

Mit „runmqckm“:

```
runmqckm -cert -delete -db filename -pw password -label label
```

runmqakm wird verwendet:

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

Dabei gilt:

-db *filename*	ist der vollständig qualifizierte Dateiname einer CMS-Schlüsseldatenbank.
-pw *password*	ist das Kennwort für die CMS-Schlüsseldatenbank.
-label *label*	ist der Kennsatz, der dem persönlichen Zertifikat zugeordnet ist.

`-fips`	Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente im FIPS-Modus nicht initialisiert wird, schlägt der Befehl **runmqakm** fehl.

ALW *Sichere Kennwörter für den Schutz des Schlüsselrepositorys unter AIX, Linux, and Windows generieren*

Sie können mit dem Befehl **runmqakm** (GSKCapiCmd) starke Kennwörter für den Schutz von Schlüsselrepositorys generieren.

Sie können den Befehl **runmqakm** mit den folgenden Parametern verwenden, um ein starkes Kennwort zu generieren:

```
runmqakm -random -create -length 14 -strong -fips
```

Wenn Sie das generierte Kennwort im Parameter **-pw** von nachfolgenden Zertifikatverwaltungsbefehlen verwenden, müssen Sie das Kennwort immer in doppelte Anführungszeichen setzen. Auf AIX and Linux-Systemen müssen Sie außerdem einen Backslash als Escapezeichen für die folgenden Zeichen verwenden, wenn sie in der Kennwortzeichenfolge vorkommen:

```
!  \  "  '
```

Wenn Sie das Kennwort als Antwort auf eine Eingabeaufforderung der **runmqckm**-, **runmqakm**- oder **strmqikm**-GUI eingeben, ist es nicht erforderlich, das Kennwort in Anführungszeichen zu setzen oder mit Escapezeichen zu versehen. Dies ist nicht erforderlich, da die Betriebssystemshell den Dateneintrag in diesen Fällen nicht beeinflusst.

ALW *Verschlüsselungshardware unter AIX, Linux, and Windows konfigurieren*

Sie können Verschlüsselungshardware für einen WS-Manager oder Client auf verschiedene Arten konfigurieren.

Sie können Verschlüsselungshardware für einen Warteschlangenmanager unter AIX, Linux, and Windows mit einer der folgenden Methoden konfigurieren:

- Verwenden Sie den MQSC-Befehl ALTER QMGR mit dem Parameter SSLCRYP (siehe Beschreibung in ALTER QMGR).
- Verwenden Sie IBM MQ Explorer, um die Verschlüsselungshardware auf Ihrem AIX, Linux, and Windows-System zu konfigurieren. Weitere Informationen finden Sie in der Onlinehilfe.

Sie können Verschlüsselungshardware für einen IBM MQ-Client unter AIX, Linux, and Windows konfigurieren, indem Sie eine der folgenden Methoden verwenden:

- Legen Sie die Umgebungsvariable MQSSLCRYP fest. Die zulässigen Werte für MQSSLCRYP sind dieselben wie für den Parameter SSLCRYP, wie im Abschnitt ALTER QMGR beschrieben.

  Wenn Sie die GSK_PKCS11 -Version des Parameters SSLCRYP verwenden, muss die PKCS #11 -Tokenbezeichnung mit der Bezeichnung übereinstimmen, mit der Sie Ihre Hardware konfiguriert haben.
- Legen Sie das Attribut SSLCryptographicHardware in der SSL-Zeilengruppe der IBM MQ client-Konfigurationsdatei fest. Die zulässigen Werte sind dieselben wie für den Parameter „SSLCRYP“, wie im Abschnitt ALTER QMGR beschrieben.

  Wenn Sie die GSK_PKCS11 -Version des Parameters SSLCRYP verwenden, muss die PKCS #11 -Tokenbezeichnung mit der Bezeichnung übereinstimmen, mit der Sie Ihre Hardware konfiguriert haben.
- Setzen Sie das Feld **CryptoHardware** der SSL-Konfigurationsoptionsstruktur (MQSCO) in einem MQCONNX-Aufruf. Weitere Informationen finden Sie im Abschnitt Übersicht für MQSCO .

Achtung: V 9.2.3 Wenn Sie die Konfiguration für die Verschlüsselungshardware über die Umgebungsvariable „MQSSLCRYP“ oder das Attribut **SSLCryptoHardware** angeben, müssen Sie das

Kennwort vor dem Speichern schützen. Weitere Informationen finden Sie unter „IBM MQ-Clients mit Verschlüsselungshardware” auf Seite 607.

Wenn Sie Verschlüsselungshardware konfiguriert haben, die die PKCS #11-Schnittstelle mit einer dieser Methoden verwendet, müssen Sie das persönliche Zertifikat für die Verwendung auf Ihren Kanälen in der Schlüsseldatenbankdatei für das verschlüsselte Token speichern, das Sie konfiguriert haben. Dieser Vorgang wird im Abschnitt „Zertifikate auf PKCS #11-Hardware verwalten” auf Seite 337 beschrieben.

ALW *Zertifikate auf PKCS #11-Hardware verwalten*

Sie können digitale Zertifikate auf Verschlüsselungshardware verwalten, die die PKCS #11-Schnittstelle unterstützt.

Informationen zu diesem Vorgang

Sie müssen eine Schlüsseldatenbank zur Vorbereitung der IBM MQ-Umgebung erstellen, selbst wenn Sie die Zertifikate der Zertifizierungsstelle nicht darin speichern möchten, sondern alle Zertifikate in Ihrer Verschlüsselungshardware speichern werden. Eine Schlüsseldatenbank ist erforderlich, damit der Warteschlangenmanager in ihrem Feld SSLKEYR referenist, oder dass die Clientanwendung in der Umgebungsvariablen MQSSLKEYR referenist. Diese Schlüsseldatenbank ist auch erforderlich, wenn Sie eine Zertifikatsanforderung erstellen.

Sie erstellen die Schlüsseldatenbank entweder über die Befehlszeile oder über die Benutzerschnittstelle von **strmqikm** (iKeyman).

Vorgehensweise

Erstellen Sie über die Befehlszeile eine Schlüsseldatenbank.

1. Führen Sie einen der folgenden Befehle aus:
   - Verwendung von **runmqckm**:

     ```
     runmqckm -keydb -create -db filename -pw password -type cms -stash
     ```

   - Verwendung von **runmqakm**:

     ```
     runmqakm -keydb -create -db filename -pw password -type cms
     -stash -fips -strong
     ```

   Dabei gilt:

   -db *Dateiname*
   Gibt den vollständig qualifizierten Dateinamen einer CMS-Schlüsseldatenbank an und muss eine Dateierweiterung von `.kdb` haben.

   -pw *password*
   Gibt das Kennwort für die CMS-Schlüsseldatenbank an.

   -type *cms*
   Gibt den Typ der Datenbank an. (Für IBM MQ muss `cms` angegeben werden.)

   -stash
   Speichert das Kennwort der Schlüsseldatenbank in einer Datei.

   -fips
   Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 validiert sind. Wenn die ICC-Komponente nicht im FIPS-Modus initialisiert wird, schlägt der Befehl **runmqakm** fehl.

   -stark
   Überprüft, ob das eingegebene Kennwort die Mindestvoraussetzungen für die Kennwortsicherheit erfüllt. Die Mindestvoraussetzungen für ein Kennwort lauten wie folgt:

   - Das Kennwort muss eine Mindestlänge von 14 Zeichen haben.

- Das Kennwort muss mindestens ein Kleinbuchstaben, ein Großbuchstaben und eine Ziffer oder ein Sonderzeichen enthalten. Zu den Sonderzeichen gehören der Stern (*), das Dollarzeichen ($), das Nummernzeichen (#) und das Prozentzeichen (%). Ein Leerzeichen wird als Sonderzeichen klassifiziert.
- Jedes Zeichen kann maximal drei Mal in einem Kennwort vorkommen.
- Es können maximal zwei aufeinanderfolgende Zeichen im Kennwort identisch sein.
- Alle Zeichen sind im Standard-ASCII-Zeichensatz für druckbare Zeichen im Bereich von 0x20 bis 0x7E enthalten.

Alternativ können Sie eine Schlüsseldatenbank über die Benutzerschnittstelle von **strmqikm** (iKeyman) erstellen.

2. Melden Sie sich auf AIX and Linux-Systemen als Rootbenutzer an. Melden Sie sich auf Windows-Systemen als Administrator oder Mitglied der Gruppe MQM an.
3. Öffnen Sie die Java-Sicherheitseigenschaftendatei `java.security`.
   - Auf AIX and Linux-Systemen befindet sich die Java-Sicherheitseigenschaftendatei im Unterverzeichnis `java/jre64/jre/lib/security` des IBM MQ-Installationsverzeichnisses.
   - Auf Windows-Systemen befindet sich die Java-Sicherheitseigenschaftendatei im Unterverzeichnis `java\jre\lib\security` des IBM MQ-Installationsverzeichnisses.

   Fügen Sie den Sicherheitsprovider IBMPKCS11Impl hinzu, falls er noch nicht in der Datei vorhanden ist. Sie können dazu beispielsweise die folgende Zeile hinzufügen:

   ```
   security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
   ```
4. Starten Sie die Benutzerschnittstelle, indem Sie den Befehl **strmqikm** ausführen.
5. Klicken Sie auf **Schlüsseldatenbankdatei** > **Öffnen**.
6. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **PKCS11Direct** aus.
7. Geben Sie im Feld **File Name** den Namen des Moduls für die Verwaltung Ihrer Verschlüsselungshardware ein, z. B. `PKCS11_API.so`.

   Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.
8. Geben Sie in das Feld **Position** den Pfad ein:
   - Auf AIX and Linux-Systemen kann dies beispielsweise `/usr/lib/pksc11` sein.
   - Auf Windows-Systemen können Sie den Bibliotheksnamen eingeben, z. B. `cryptoki`.

   Klicken Sie auf **OK**. Das Fenster 'Open Cryptographic Token' wird geöffnet.
9. Wählen Sie die Tokenbezeichnung für die Verschlüsselungseinheit aus, unter der Sie die Zertifikate speichern möchten.
10. Geben Sie im Feld **Cryptographic Token Password** das Kennwort ein, das Sie bei der Konfiguration der Verschlüsselungshardware festgelegt haben.
11. Wenn Ihre Verschlüsselungshardware die Unterzeichnerzertifikate speichern kann, die zum Anfordern oder Importieren persönlicher Zertifikate erforderlich sind, deaktivieren Sie die beiden Kontrollkästchen für die sekundäre Schlüsseldatenbank, und fahren Sie mit Schritt „15” auf Seite 339 fort.

    Wenn Sie eine sekundäre CMS-Schlüsseldatenbank benötigen, um die Unterzeichnerzertifikate zu speichern, wählen Sie entweder **Vorhandene Sekundärschlüsseldatenbankdatei öffnen** oder **Neue Sekundärschlüsseldatenbankdatei erstellen** aus.
12. Geben Sie in das Feld **Dateiname** einen Dateinamen ein. Dieses Feld enthält bereits den Text `key.kdb`. Wenn Ihr Stammname `key` ist, lassen Sie dieses Feld unverändert. Wenn Sie einen anderen Stammnamen angegeben haben, ersetzen Sie `key` durch Ihren Stammnamen. Sie dürfen das Suffix `.kdb` nicht ändern.

13. Geben Sie in das Feld **Position** den Pfad ein, z. B.:
    - Für einen Warteschlangenmanager: `/var/mqm/qmgrs/QM1/ssl`
    - Für einen IBM MQ MQI client: `/var/mqm/ssl`

    Klicken Sie auf **OK**. Das Fenster "Password Prompt" wird geöffnet.
14. Geben Sie ein Kennwort ein.

    Wenn Sie in Schritt „11" auf Seite 338 die Option **Vorhandene Sekundärschlüsseldatenbankdatei öffnen** ausgewählt haben, geben Sie im Feld **Kennwort** ein Kennwort ein.

    Wenn Sie in Schritt „11" auf Seite 338 **Neue sekundäre Schlüsseldatenbankdatei erstellen** ausgewählt haben, führen Sie die folgenden Unterschritte aus:

    a) Geben Sie ein Kennwort in das Feld **Kennwort** ein, und geben Sie es erneut in das Feld **Kennwort bestätigen** ein.

    b) Wählen Sie **Kennwort in einer Datei speichern** aus. Wenn Sie das Kennwort nicht verstellen, schlagen Versuche zum Starten von TLS-Kanälen fehl, da sie das Kennwort, das für den Zugriff auf die Schlüsseldatenbankdatei erforderlich ist, nicht abrufen können.

    c) Klicken Sie auf **OK**. Es wird ein Fenster angezeigt, in dem bestätigt wird, dass sich das Kennwort in der Datei `key.sth` befindet (sofern Sie keinen anderen Stammnamen angegeben haben).
15. Klicken Sie auf **OK**.

    Der Inhaltsrahmen der Schlüsseldatenbank wird angezeigt.

ALW *Anfordern eines persönlichen Zertifikats für Ihre PKCS #11-Hardware*

Verwenden Sie diese Vorgehensweise für einen Warteschlangenmanager oder einen IBM MQ MQI client, um ein persönliches Zertifikat für Ihre Verschlüsselungshardware anzufordern.

Informationen zu diesem Vorgang

In dieser Task wird beschrieben, wie Sie die **strmqikm**-Benutzerschnittstelle verwenden, um ein persönliches Zertifikat anzufordern. Wenn Sie die Befehlszeilenschnittstelle verwenden, lesen Sie die Informationen unter „Verwenden der Befehlszeile" auf Seite 320.

Anmerkung: IBM MQ unterstützt keine SHA-3- oder SHA-5-Algorithmen. Sie können die Namen der digitalen Signaturalgorithmen SHA384WithRSA und SHA512WithRSA verwenden, da beide Algorithmen zu Mitgliedern der SHA-2-Familie gehören.

Die Namen der digitalen Signaturalgorithmen SHA3WithRSA und SHA5WithRSA werden nicht weiter unterstützt, da sie eine abgekürzte Form von SHA384WithRSA bzw. SHA512WithRSA sind.

Vorgehensweise

Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat von der Benutzerschnittstelle **strmqikm** (iKeyman) anzufordern:

1. Führen Sie die Schritte aus, um mit der Verschlüsselungshardware zu arbeiten. Weitere Informationen finden Sie in „Zertifikate auf PKCS #11-Hardware verwalten" auf Seite 337.
2. Klicken Sie im Menü **Erstellen** auf **Neue Zertifikatsanforderung** .

   Das Fenster "Neuen Schlüssel-und Zertifikatsanforderung erstellen" wird geöffnet.
3. Geben Sie in das Feld **Schlüsselkennsatz** die Zertifikatsbezeichnung ein.

   Bei der Bezeichnung handelt es sich um den Wert des Attributs **CERTLABL**, falls dieses festgelegt ist, oder um den Standardwert `ibmwebspheremq`, an den der Name des Warteschlangenmanagers oder die Anmeldebenutzer-ID des IBM MQ MQI clients (jeweils in Kleinschreibung) angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .
4. Wählen Sie die erforderliche **Schlüsselgröße** und den **Signaturalgorithmus** aus.

5. Geben Sie Werte für **Common Name** und **Organization** ein, und wählen Sie ein **Land** aus. Geben Sie für die verbleibenden optionalen Felder entweder die Standardwerte an, oder geben Sie neue Werte ein oder wählen Sie neue Werte aus.

   Beachten Sie, dass Sie im Feld **Organisationseinheit** nur einen Namen angeben können. Weitere Informationen zu diesen Feldern finden Sie unter „Definierte Namen" auf Seite 14.
6. Geben Sie im Feld **Geben Sie den Namen einer Datei ein, in die das Zertifikatsanforderung gespeichert werden soll** entweder den Standardwert `certreq.arm` ein, oder geben Sie einen neuen Wert mit einem vollständigen Pfad ein.
7. Klicken Sie auf **OK**.

   Ein Bestätigungsfenster wird geöffnet.
8. Klicken Sie auf **OK**.

   In der Liste **Persönliche Zertifikatsanforderungen** wird die Bezeichnung der neuen persönlichen Zertifikatsanforderung angezeigt, die Sie erstellt haben. Die Zertifikatsanforderung wird in der unter Punkt „6" auf Seite 340 ausgewählten Datei gespeichert.
9. Fordern Sie das neue persönliche Zertifikat an, indem Sie die Datei an eine Zertifizierungsstelle (CA) senden oder indem Sie die Datei in das Anforderungsformular auf der Website für die CA kopieren.

ALW *Persönliches Zertifikat in Ihrer PKCS #11-Hardware empfangen*

Verwenden Sie dieses Verfahren für einen Warteschlangenmanager oder einen IBM MQ MQI client, um ein persönliches Zertifikat in Ihrer Verschlüsselungshardware zu empfangen.

Vorbereitende Schritte

Fügen Sie das CA-Zertifikat der Zertifizierungsstelle hinzu, von der das persönliche Zertifikat signiert wurde. Fügen Sie das Zertifikat in die Verschlüsselungshardware oder die sekundäre CMS-Schlüsseldatenbank ein. Führen Sie diesen Vorgang aus, bevor Sie das signierte Zertifikat in der Verschlüsselungshardware empfangen. Zum Hinzufügen eines CA-Zertifikat zu einem Schlüsselring folgen Sie den Anweisungen in „CA-Zertifikat oder öffentlichen Teil eines selbst signierten Zertifikats unter AIX, Linux, and Windows einem Schlüsselrepository hinzufügen" auf Seite 327.

Prozedur

- Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat mit der Benutzerschnittstelle **strmqikm** (iKeyman) zu erhalten:
  a) Führen Sie die Schritte aus, um mit der Verschlüsselungshardware zu arbeiten. Weitere Informationen finden Sie in „Zertifikate auf PKCS #11-Hardware verwalten" auf Seite 337.
  b) Klicken Sie auf **Empfangen** . Das Fenster 'Receive Certificate from a File' (Zertifikat aus einer Datei empfangen) wird angezeigt.
  c) Geben Sie den Namen und die Position der Zertifikatsdatei für das neue persönliche Zertifikat ein, oder klicken Sie auf **Durchsuchen** , um den Namen und die Position auszuwählen.
  d) Klicken Sie auf **OK**. Wenn Sie bereits ein persönliches Zertifikat in Ihrer Schlüsseldatenbank haben, wird ein Fenster geöffnet, in dem Sie gefragt werden, ob Sie den Schlüssel, den Sie als Standardschlüssel hinzufügen möchten, in der Datenbank festlegen möchten.
  e) Klicken Sie auf **Ja** oder **Nein**. Das Fenster "Enter a Label" wird geöffnet.
  f) Klicken Sie auf **OK**. In der Liste **Persönliche Zertifikate** wird die Bezeichnung des neuen persönlichen Zertifikats angezeigt, das Sie hinzugefügt haben. Dieser Kennsatz wird durch Hinzufügen des Kennsatzes des Verschlüsselungstokens vor dem von Ihnen angegebenen Kennsatz gebildet.
- Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat mit dem Befehl **runmqakm** (GSKCapiCmd) zu empfangen:
  a) Öffnen Sie ein Befehlsfenster, das für Ihre Umgebung konfiguriert ist.
  b) Empfangen Sie ein persönliches Zertifikat mit dem Befehl **runmqakm** (GSKCapiCmd):

```
runmqakm -cert -receive -file filename -crypto module_name
```

```
          -tokenlabel hardware_token -pw hardware_password
          -format cert_format -fips
          -secondaryDB filename -secondaryDBpw password
```

Dabei gilt:

-file *Dateiname*
Gibt den vollständig qualifizierten Dateinamen der Datei an, die das persönliche Zertifikat enthält.

-crypto *Modulname*
Gibt den vollständig qualifizierten Namen der PKCS #11-Bibliothek an, die mit der Verschlüsselungshardware geliefert wird.

-tokenlabel *Hardware-Token*
Gibt die Tokenbezeichnung für die PKCS #11-Verschlüsselungseinheit an.

-pw *Hardware-Kennwort*
Gibt das Kennwort für den Zugriff auf die Verschlüsselungshardware an.

-format *Zertifikatsformat*
Gibt das Format des Zertifikats an. Der Wert kann `ascii` für Base64-encoded ASCII oder `binary` für binäre DER-Daten sein. Der Standardwert ist ASCII.

-fips
Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 validiert sind. Wenn die ICC-Komponente im FIPS-Modus nicht initialisiert wird, schlägt der Befehl **runmqakm** fehl.

-secondaryDB *Dateiname*
Gibt den vollständig qualifizierten Dateinamen der CMS-Schlüsseldatenbank an.

-secondaryDBpw *Kennwort*
Gibt das Kennwort für die CMS-Schlüsseldatenbank an.

MQ Appliance Mit SSL/TLS unter IBM MQ Appliance arbeiten

Für IBM MQ Appliance wird Transport Layer Security (TLS) unterstützt.

IBM MQ Appliance verfügt über eindeutige Befehle zum Verwalten von Zertifikaten. Weitere Informationen zur Zertifikatsverwaltung finden Sie in der IBM MQ Appliance-Dokumentation unter TLS certificate management

z/OS Mit SSL/TLS unter z/OS arbeiten

In diesen Informationen wird beschrieben, wie Sie Transport Layer Security (TLS) unter z/OS einrichten und verwenden.

Jedes Thema enthält Beispiele für die Ausführung der einzelnen Tasks mithilfe von RACF. Sie können ähnliche Tasks mit den anderen externen Sicherheitsmanagern ausführen.

Unter z/OS müssen Sie außerdem die Anzahl der Serversubtasks festlegen, die jeder Warteschlangenmanager für die Verarbeitung von TLS-Aufrufen verwendet, wie unter „Parameter SSLTASKS unter z/OS festlegen” auf Seite 342 beschrieben.

Die TLS-Unterstützung für z/OS ist ein integraler Bestandteil des Betriebssystems und wird als *System SSL* bezeichnet. System SSL ist Teil des Cryptographic Services Base-Elements von z/OS. Die Cryptographic Services Base-Member werden im *pdsname* installiert. Partitionierte Datei SIEALNKE (PDS) . Wenn Sie System SSL installieren, stellen Sie sicher, dass Sie die entsprechenden Optionen auswählen, um die erforderlichen CipherSpecs zur Verfügung zu stellen.

z/OS *Zusätzliche Benutzer-ID-Anforderungen für TLS unter z/OS*

In diesem Informationen werden die zusätzlichen Anforderungen beschrieben, die Ihre Benutzer-ID für die Einrichtung und Arbeit mit TLS unter z/OS benötigt.

Stellen Sie sicher, dass alle erforderlichen HIPER-Aktualisierungen (HIPER-High Impact oder Pervasive) auf Ihrem System vorhanden sind.

Stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllt haben:

- Die Benutzer-ID *ssidCHIN* ist in RACF korrekt definiert und die Benutzer-ID *ssidCHIN* hat Lesezugriff (READ) auf die folgenden Profile:
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

  Diese Variablen sind in der RACF-Klasse FACILITY definiert.
- Die *ssidCHIN* -Benutzer-ID ist der Eigner des Schlüsselrings.
- Das persönliche Zertifikat des Warteschlangenmanagers wird, wenn es mit dem Befehl RACDCERT erstellt wird, mit einer Benutzer-ID des Zertifikatstyps erstellt, die auch mit der *ssidCHIN* -Benutzer-ID identisch ist.
- Der Kanalinitiator wird erneut gestartet, oder der Befehl **REFRESH SECURITY TYPE (SSL)** wird ausgegeben, um alle Änderungen zu übernehmen, die Sie am Schlüsselring vornehmen.
- Die Prozedur für den IBM MQ-Kanalinitiator kann über die Linkliste, LPA oder eine STEPLIB-Datendefinitionsanweisung auf die SSL-Laufzeitbibliothek *pdsname*.SIEALNKE des Systems zugreifen. Diese Bibliothek muss APF-berechtigt sein.
- Die Benutzer-ID, unter deren Berechtigung der Kanalinitiator ausgeführt wird, ist für die Verwendung von z/OS UNIX System Services (z/OS UNIX) konfiguriert, wie in der Dokumentation zur Planung von z/OS UNIX System Services beschrieben.

  Benutzer, die nicht möchten, dass der Kanalinitiator z/OS UNIX mit dem Segment 'guest/default UID' und 'OMVS' aufruft, müssen nur ein neues OMVS-Segment modellieren, das auf dem Standardsegment basiert, da der Kanalinitiator keine speziellen Berechtigungen erfordert und nicht innerhalb von UNIX als Superuser ausgeführt wird.

z/OS *Parameter SSLTASKS unter z/OS festlegen*

Verwenden Sie den Befehl ALTER QMGR, um die Anzahl der Server-Subtasks für die Verarbeitung von TLS-Aufrufen festzulegen.

Um TLS-Kanäle verwenden zu können, müssen Sie sicherstellen, dass mindestens zwei Serversubtasks vorhanden sind, indem Sie den Parameter SSLTASKS mit dem Befehl ALTER QMGR definieren. Beispiel:

```
ALTER QMGR SSLTASKS(5)
```

Um Probleme mit der Speicherzuordnung zu vermeiden, setzen Sie das Attribut SSLTASKS nicht auf einen Wert größer als acht in einer Umgebung, in der keine CRL-Prüfung (Certificate Revocation List, Zertifikatswiderrufsliste) vorhanden ist.

Wenn die CRL-Prüfung verwendet wird, wird eine SSLTASK von dem betreffenden Kanal für die Dauer dieser Prüfung angehalten. Hierbei kann es sich um einen erheblichen Zeitraum handeln, während dem der relevante LDAP-Server kontaktiert wird, da es sich bei jedem SSLTASK um einen Tasksteuerblock für z/OS handelt.

Sie müssen den Kanalinitiator erneut starten, wenn Sie den Wert des Attributs SSLTASKS ändern.

z/OS *Schlüsselrepository unter z/OS einrichten*

Richten Sie ein Schlüsselrepository an beiden Enden der Verbindung ein. Ordnen Sie jedem Schlüsselrepository den zugehörigen Warteschlangenmanager zu.

Für eine TLS-Verbindung ist an jedem Ende der Verbindung ein *Schlüsselrepository* erforderlich. Jeder WS-Manager muss Zugriff auf ein Schlüsselrepository haben. Verwenden Sie den Parameter SSLKEYR im Befehl ALTER QMGR, um einem WS-Manager ein Schlüsselrepository zuzuordnen. Weitere Informationen finden Sie im Abschnitt „Das SSL/TLS-Schlüsselrepository” auf Seite 27.

Unter z/OS werden digitale Zertifikate in einem *Schlüsselring* gespeichert, der von Ihrem externen Sicherheitsmanager (ESM) verwaltet wird. Diese digitalen Zertifikate weisen Beschriftungen auf, die das Zertifikat einem Warteschlangenmanager zuordnen. TLS verwendet diese Zertifikate zu Authentifizierungszwecken. In allen nachfolgenden Beispiel werden RACF-Befehle verwendet. Für andere ESM-Programme sind äquivalente Befehle vorhanden.

Unter z/OS verwendet IBM MQ den Wert des Attributs **CERTLABL**, falls dieser festgelegt ist, oder den Standardwert `ibmWebSphereMQ`, an den der Name des Warteschlangenmanagers angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

Der Name des Schlüsselrepositorys für einen Warteschlangenmanager ist der Name eines Schlüsselrings in Ihrer RACF-Datenbank. Sie können den Schlüsselringnamen vor oder nach der Erstellung des Schlüsselrings angeben.

Gehen Sie wie folgt vor, um einen neuen Schlüsselring für einen WS-Manager zu erstellen:

1. Stellen Sie sicher, dass Sie über die entsprechende Berechtigung zur Ausgabe des Befehls RACDCERT verfügen (weitere Informationen finden Sie im Handbuch *SecureWay Security Server RACF Command Language Reference*).
2. Geben Sie den folgenden Befehl ein:

   ```
   RACDCERT ID( userid1 ) ADDRING( ring-name )
   ```

   Dabei gilt:

   - *userid1* ist die Benutzer-ID des Kanalinitiatoradressraums oder die Benutzer-ID, die Eigner des Schlüsselrings sein wird (wenn der Schlüsselring gemeinsam genutzt wird).
   - *Ringname* ist der Name, den Sie dem Schlüsselring geben möchten. Die Länge dieses Namens kann bis zu 237 Zeichen lang sein. Bei diesem Namen muss die Groß-/Kleinschreibung beachtet werden. Geben Sie *ring-name* in Großbuchstaben an, um Probleme zu vermeiden.

z/OS *CA-Zertifikate einem Warteschlangenmanager unter z/OS zugänglich machen*

Nachdem Sie den Schlüsselring erstellt haben, schließen Sie alle relevanten CA-Zertifikate an diesen Ring an.

Wenn das CA-Zertifikat in einem Dataset enthalten ist, müssen Sie das Zertifikat zuerst mit folgendem Befehl der RACF-Datenbank hinzufügen:

```
RACDCERT ID( userid1 ) ADD( input-data-set-name ) WITHLABEL( 'My CA' )
```

Verwenden Sie anschließend den folgenden Befehl, um ein CA-Zertifikat für `My  CA` mit dem Schlüsselring zu verbinden:

```
RACDCERT ID(userid1)
CONNECT(CERTAUTH LABEL('My CA') RING(ring-name) USAGE(CERTAUTH))
```

Dabei ist *userid1* entweder die Kanalinitiatorbenutzer-ID oder der Eigner eines gemeinsam genutzten Schlüsselrings.

Weitere Informationen zu CA-Zertifikaten finden Sie unter „Digitale Zertifikate” auf Seite 12.

z/OS *Schlüsselrepository für einen Warteschlangenmanager unter z/OS ermitteln*

Verwenden Sie diese Prozedur, um die Position des Schlüsselrings Ihres WS-Managers zu ermitteln.

1. Zeigen Sie die Attribute des WS-Managers mit einem der folgenden MQSC-Befehle an:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

2. Überprüfen Sie die Befehlsausgabe auf die Position des Schlüsselrings.

z/OS *Position des Schlüsselrepositorys für einen Warteschlangenmanager unter z/OS angeben*

Wenn Sie die Position des Schlüsselrings Ihres WS-Managers angeben möchten, verwenden Sie den MQSC-Befehl ALTER QMGR, um das Schlüsselrepository-Attribut des WS-Managers festzulegen.

Beispiel:

```
ALTER QMGR SSLKEYR(CSQ1RING)
```

wenn der Schlüsselring dem Adressraum des Kanalinitiators zugeordnet ist, oder:

```
ALTER QMGR SSLKEYR(userid1/CSQ1RING)
```

wenn es sich um einen gemeinsam genutzten Schlüsselring handelt, wobei *userid1* die Benutzer-ID ist, die Eigner des Schlüsselrings ist.

z/OS *Kanalinitiator die ordnungsgemäßen Zugriffsberechtigungen unter z/OS erteilen*

Der Kanalinitiator (CHINIT) benötigt Zugriff auf das Schlüsselrepository und auf bestimmte Sicherheitsprofile.

CHINIT-Zugriff für das Lesen des Schlüsselrepositorys erteilen

Wenn die CHINIT-Benutzer-ID Eigner des Schlüsselrepositorys ist, muss diese Benutzer-ID Lesezugriff auf das Profil IRR.DIGTCERT.LISTRING in der Klasse FACILITY haben, und der Zugriff auf andere Benutzer aktualisieren. Erteilen Sie den Zugriff wie folgt mit dem Befehl PERMIT mit ACCESS (UPDATE) oder ACCESS (READ):

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID( userid ) ACCESS(UPDATE)
```

Hierbei steht *userid* für die Benutzer-ID des Adressraums des Kanalinitiators.

CHINIT-Lesezugriff auf die entsprechenden CSF* -Profile erteilen

Stellen Sie sicher, dass Ihre CHINIT-Benutzer-ID über den folgenden Befehl Lesezugriff auf die entsprechenden CSF* -Profile in der CSFSERV-Klasse hat, wenn die Hardware-Unterstützung über die Integrated Cryptographic Service Facility (ICSF) zur Verfügung gestellt wird:

```
PERMIT csf-resource CLASS(CSFSERV) ID( userid ) ACCESS(READ)
```

Hierbei steht *csf-resource* für den Namen des CSF* -Profils und *userid* für die Benutzer-ID des Adressraums des Kanalinitiators.

Wiederholen Sie diesen Befehl für jedes der folgenden CSF* -Profile:

- CSFDSG
- CSFDSV
- CSFPKD
- CSFPKE
- CSFPKI

Für Ihre CHINIT-Benutzer-ID ist möglicherweise ebenfalls Lesezugriff auf andere CSF*-Profile erforderlich. Wenn Sie beispielsweise die CipherSpec ECDHE_RSA_AES_256_GCM_SHA384 verwenden, benötigt Ihre CHINIT-Benutzer-ID ebenfalls Lesezugriff auf die folgenden CSF*-Profile:

- CSF1DVK
- CSF1GAV
- CSF1GKP
- CSF1SKE
- CSF1TRC
- CSF1TRD

Weitere Informationen finden Sie unter RACF-CSFSERV-Ressourcenanforderungen.

Wenn Ihre Zertifikatschlüssel in ICSF gespeichert sind und die Installation über die in ICSF gespeicherten Schlüssel verfügt, stellen Sie sicher, dass Ihre CHINIT-Benutzer-ID über Lesezugriff auf das Profil in der Klasse CSFKEYS verfügt, indem Sie den folgenden Befehl verwenden:

```
PERMIT IRR.DIGTCERT. userid.* CLASS(CSFKEYS) ID( userid ) ACCESS(READ)
```

Hierbei steht *userid* für die Benutzer-ID des Adressraums des Kanalinitiators.

Integrated Cryptographic Service Facility (ICSF) verwenden

Der Kanalinitiator kann ICSF verwenden, um eine zufällige Zahl zu generieren, wenn der Kennwortschutzalgorithmus zum Absetzen von Kennwörtern verwendet wird, die über Clientkanäle fließen, wenn TLS nicht verwendet wird.

Weitere Informationen finden Sie unter „Integrated Cryptographic Service Facility (ICSF) verwenden” auf Seite 283

z/OS *Zeitpunkt, an dem Änderungen an Zertifikaten oder dem Schlüsselrepository unter z/OS wirksam werden*

Änderungen werden wirksam, wenn der Kanalinitiator gestartet wird oder wenn das Repository aktualisiert wird.

Insbesondere werden Änderungen an den Zertifikaten im Schlüsselring und an dem Schlüssel-Repository-Attribut in einem der folgenden Fälle wirksam:

- Wenn der Kanalinitiator gestartet oder erneut gestartet wird.
- Wenn der Befehl REFRESH SECURITY TYPE (SSL) ausgegeben wird, um den Inhalt des Schlüsselrepositorys zu aktualisieren.

z/OS *Selbst signiertes persönliches Zertifikat unter z/OS erstellen*

Verwenden Sie diese Prozedur, um ein selbst signiertes persönliches Zertifikat zu erstellen.

1. Generieren Sie ein Zertifikat und ein öffentliches und privates Schlüsselpaar mit dem folgenden Befehl:

   ```
   RACDCERT ID(userid2) GENCERT
   SUBJECTSDN(CN('common-name')
              T('title')
              OU('organizational-unit')
              O('organization')
              L('locality')
              SP('state-or-province')
              C('country'))
   WITHLABEL('label-name')
   ```

2. Schließen Sie das Zertifikat mit dem folgenden Befehl an den Schlüsselring an:

   ```
   RACDCERT ID(userid1)
   CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
   ```

Dabei gilt:

- *userid1* ist die Benutzer-ID des Kanalinitiatoradressraums oder des Eigners des gemeinsam genutzten Schlüsselrings.
- *userid2* ist die Benutzer-ID, die dem Zertifikat zugeordnet ist, und muss die Benutzer-ID des Kanalinitiatoradressraums sein.

  userid1 und *userid2* können die gleiche ID sein.
- *Ringname* ist der unter „Schlüsselrepository unter z/OS einrichten” auf Seite 342 vergebene Name für den Schlüsselring.
- *label-name* muss entweder der Wert des Attributs IBM MQ **CERTLABL** sein, wenn er gesetzt ist, oder der Standardwert `ibmWebSphereMQ` mit dem Namen des angehängten Warteschlangenmanagers. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

z/OS *Persönliches Zertifikat unter z/OS anfordern*

Fordern Sie ein persönliches Zertifikat mithilfe von RACF an.

Wenn Sie ein persönliches Zertifikat anfordern möchten, verwenden Sie RACF folgendermaßen:

1. Erstellen Sie wie in „Selbst signiertes persönliches Zertifikat unter z/OS erstellen” auf Seite 345 beschrieben ein selbst signiertes persönliches Zertifikat. Dieses Zertifikat stellt die Anforderung mit den Attributwerten für den definierten Namen (DN) bereit.
2. Erstellen Sie mit dem folgenden Befehl eine PKCS #10 Base64-verschlüsselte Zertifikatsanforderung, die in eine Datei geschrieben wurde:

   ```
   RACDCERT ID(userid2) GENREQ(LABEL(' label_name ')) DSN(' output_data_set_name ')
   ```

   Dabei gilt Folgendes:

   - *userid2* ist die Benutzer-ID, die dem Zertifikat zugeordnet ist, und muss die Benutzer-ID des Kanalinitiatoradressraums sein.
   - *label_name* ist die Bezeichnung, die beim Erstellen des selbst signierten Zertifikats verwendet wird.

   Details finden Sie im Abschnitt „Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen” auf Seite 29.
3. Senden Sie die Datei an eine Zertifizierungsinstanz (CA), um ein neues persönliches Zertifikat anzufordern.
4. Wenn das signierte Zertifikat von der Zertifizierungsstelle an Sie zurückgegeben wird, fügen Sie das Zertifikat wieder in die RACF-Datenbank ein und verwenden Sie dabei die ursprüngliche Bezeichnung (siehe „Persönliche Zertifikate in ein Schlüsselrepository unter z/OS hinzufügen” auf Seite 347=.

z/OS *Signiertes persönliches Zertifikat für RACF erstellen*

RACF kann als Zertifizierungsstelle dienen und eigene CA-Zertifikate ausgeben.

In diesem Abschnitt wird der Begriff *Unterzeichnerzertifikat* verwendet, um ein CA-Zertifikat anzugeben, das von RACF ausgestellt wurde.

Der private Schlüssel für das Unterzeichnerzertifikat muss sich in der RACF-Datenbank befinden, bevor Sie die folgenden Schritte ausführen:

1. Mit dem folgenden Befehl generieren Sie mithilfe des Unterzeichnerzertifikats in Ihrer RACF-Datenbank ein persönliches Zertifikat, das von RACF signiert ist:

   ```
   RACDCERT ID(userid2) GENCERT
   SUBJECTSDN(CN('common-name')
              T('title')
              OU('organizational-unit')
              O('organization')
              L('locality')
              SP('state-or-province')
   ```

```
           C('country'))
WITHLABEL('label-name')
SIGNWITH(CERTAUTH LABEL('signer-label'))
```

2. Schließen Sie das Zertifikat mit dem folgenden Befehl an den Schlüsselring an:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

Dabei gilt:

- *userid1* ist die Benutzer-ID des Kanalinitiatoradressraums oder des Eigners des gemeinsam genutzten Schlüsselrings.
- *userid2* ist die Benutzer-ID, die dem Zertifikat zugeordnet ist, und muss die Benutzer-ID des Kanalinitiatoradressraums sein.

  userid1 und *userid2* können die gleiche ID sein.
- *Ringname* ist der unter „Schlüsselrepository unter z/OS einrichten” auf Seite 342 vergebene Name für den Schlüsselring.
- *label-name* muss entweder der Wert des Attributs IBM MQ **CERTLABL** sein, wenn er gesetzt ist, oder der Standardwert `ibmWebSphereMQ` an den der Name des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange angehängt wird. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .
- *signer-label* ist die Bezeichnung Ihres eigenen Unterzeichnerzertifikats.

z/OS *Persönliche Zertifikate in ein Schlüsselrepository unter z/OS hinzufügen*

Verwenden Sie diese Prozedur zum Hinzufügen oder Importieren eines persönlichen Zertifikats zu einem Schlüsselring.

Nachdem die Zertifizierungsstelle Ihnen ein neues persönliches Zertifikat gesendet hat, fügen Sie sie mit der folgenden Prozedur zum Schlüsselring hinzu:

1. Fügen Sie das Zertifikat mit folgendem Befehl der RACF-Datenbank hinzu:

```
RACDCERT ID( userid2 ) ADD( input-data-set-name ) WITHLABEL(' label-name ')
```

2. Schließen Sie das Zertifikat mit dem folgenden Befehl an den Schlüsselring an:

```
RACDCERT ID( userid1 )
CONNECT(ID( userid2 ) LABEL(' label-name ') RING( ring-name ) USAGE(PERSONAL))
```

Dabei gilt:

- *userid1* ist die Benutzer-ID des Kanalinitiatoradressraums oder des Eigners des gemeinsam genutzten Schlüsselrings.
- *userid2* ist die Benutzer-ID, die dem Zertifikat zugeordnet ist, und muss die Benutzer-ID des Kanalinitiatoradressraums sein.
- *Ringname* ist der unter „Schlüsselrepository unter z/OS einrichten” auf Seite 342 vergebene Name für den Schlüsselring.
- *Name des Eingabedatensatzes* ist der Name des Datensatzes, der das signierte CA-Zertifikat enthält. Die Dateigruppe muss katalogisiert sein und darf keine PDS oder ein Member einer PDS sein. Das von RACDCERT erwartete Satzformat (RECFM) ist VB. RACDCERT ordnet die Datei dynamisch zu und öffnet den Datensatz und liest das Zertifikat als binäre Daten aus.
- *label-name* ist der Kennsatzname, der beim Erstellen der ursprünglichen Anforderung verwendet wurde. Dabei muss es sich um den Wert des Attributs IBM MQ **CERTLABL** handeln, wenn er festgelegt ist, oder um den Standardwert `ibmWebSphereMQ`, an den der Name des Warteschlangenmanagers

oder der Gruppe mit gemeinsamer Warteschlange angehängt ist. Weitere Informationen finden Sie im Abschnitt Digital Certificate Labels .

z/OS *Persönliches Zertifikat aus einem Schlüsselrepository unter z/OS exportieren*

Exportieren Sie das Zertifikat mit dem Befehl RACDCERT.

Verwenden Sie auf dem System, von dem aus Sie das Zertifikat exportieren möchten, den folgenden Befehl:

```
RACDCERT ID(userid2) EXPORT(LABEL('label-name'))
DSN(output-data-set-name) FORMAT(CERTB64)
```

Dabei gilt:

- *userid2* ist die Benutzer-ID, unter der das Zertifikat zum Schlüsselring hinzugefügt wurde.
- *label-name* ist die Bezeichnung des Zertifikats, das Sie extrahieren möchten.
- *Ausgabedatei-Name* ist der Datensatz, in den das Zertifikat gestellt wird.
- CERTB64 ist ein DER-codiertes X.509-Zertifikat, das im Base64-Format vorliegt. Sie können ein anderes Format auswählen, z. B.:

  CERTDER
  DER-codiertes X.509-Zertifikat im Binärformat

  PKCS12B64
  PKCS#12-Zertifikat im Base64-Format

  PKCS12DER
  PKCS#12-Zertifikat im Binärformat

z/OS *Persönliches Zertifikat aus einem Schlüsselrepository unter z/OS löschen*

Löschen Sie ein persönliches Zertifikat mit dem Befehl RACDCERT.

Bevor Sie ein persönliches Zertifikat löschen, können Sie eine Kopie davon speichern. Eine Beschreibung, wie Sie Ihr persönliches Zertifikat vor dem Löschen in einen Datensatz kopieren, finden Sie unter „Persönliches Zertifikat aus einem Schlüsselrepository unter z/OS exportieren” auf Seite 348. Verwenden Sie dann den folgenden Befehl, um Ihr persönliches Zertifikat zu löschen:

```
RACDCERT ID( userid2 ) DELETE(LABEL(' label-name '))
```

Dabei gilt:

- *userid2* ist die Benutzer-ID, unter der das Zertifikat zum Schlüsselring hinzugefügt wurde.
- *kennsatzname* ist der Name des Zertifikats, das gelöscht werden soll.

z/OS *Persönliches Zertifikat in einem Schlüsselrepository unter z/OS umbenennen*

Benennen Sie ein Zertifikat mit dem Befehl RACDCERT um.

Wenn Sie nicht möchten, dass ein Zertifikat mit einer bestimmten Bezeichnung gefunden wird, aber es nicht löschen möchten, können Sie es mit dem folgenden Befehl vorübergehend umbenennen:

```
RACDCERT ID( userid2 ) LABEL(' label-name ') NEWLABEL(' new-label-name ')
```

Dabei gilt:

- *userid2* ist die Benutzer-ID, unter der das Zertifikat zum Schlüsselring hinzugefügt wurde.
- *Kennsatzname* ist der Name des Zertifikats, das Sie umbenennen möchten.

- *new-label-name* ist der neue Name des Zertifikats.

Dies kann beim Testen der TLS-Clientauthentifizierung nützlich sein.

z/OS *Benutzer-ID unter z/OS einem digitalen Zertifikat zuordnen*

IBM MQ kann eine Benutzer-ID, die einem RACF-Zertifikat zugeordnet ist, als Kanalbenutzer-ID verwenden. Ordnen Sie eine Benutzer-ID einem Zertifikat zu, indem Sie es unter dieser Benutzer-ID installieren oder einen Zertifikatnamensfilter verwenden.

Die in diesem Thema beschriebene Methode ist eine Alternative zu der plattformunabhängigen Methode für die Zuordnung einer Benutzer-ID zu einem digitalen Zertifikat, das Kanalauthentifizierungsdatensätze verwendet. Weitere Informationen zu Kanalauthentifizierungsdatensätzen finden Sie unter „Kanalauthentifizierungssätze” auf Seite 54.

Wenn eine Entität an einem Ende eines TLS-Kanals ein Zertifikat von einer fernen Verbindung erhält, so fragt diese Entität bei RACF an, ob dem Zertifikat eine Benutzer-ID zugeordnet ist. Die Entität verwendet diese Benutzer-ID als Kanalbenutzer-ID. Wenn dem Zertifikat keine Benutzer-ID zugeordnet ist, verwendet die Entität die Benutzer-ID, unter der der Kanalinitiator ausgeführt wird.

Ordnen Sie eine Benutzer-ID auf eine der folgenden Arten zu einem Zertifikat zu:

- Installieren Sie dieses Zertifikat in der RACF-Datenbank unter der Benutzer-ID, der es zugeordnet werden soll, wie unter „Persönliche Zertifikate in ein Schlüsselrepository unter z/OS hinzufügen” auf Seite 347 beschrieben.
- Ordnen Sie wie in „Zertifikatsnamensfilter unter z/OS einrichten” auf Seite 349 beschrieben den definierten Namen des Besitzers oder Ausstellers des Zertifikats mithilfe eines Zertifikatsnamensfilters (Certificate Name Filter; CNF) der Benutzer-ID zu.

z/OS *Zertifikatsnamensfilter unter z/OS einrichten*

Verwenden Sie den Befehl RACDCERT, um einen Zertifikatnamensfilter (CNF) zu definieren, der einen definierten Namen (Distinguished Name) einer Benutzer-ID zuordnet.

Führen Sie die folgenden Schritte aus, um ein CNF einzurichten.

1. Aktivieren Sie CNF-Funktionen mit dem folgenden Befehl. Um dies zu tun, benötigen Sie die Aktualisierungsberechtigung für die Klasse DIGTNMAP.

   ```
   SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
   ```

2. Definieren Sie das CNF. Beispiel:

   ```
   RACDCERT ID(USER1) MAP WITHLABEL('filter1') TRUST
   SDNFILTER('O=IBM.C=UK') IDNFILTER('O=ExampleCA.L=Internet')
   ```

   Dabei steht USER1 für die Benutzer-ID, die verwendet werden soll, wenn:

   - Der DN des Subjekt hat eine Organisation von `IBM` und ein Land von UK.
   - Der DN des Ausstellers hat eine Organisation von `ExampleCA` und eine Lokalität von `Internet`.

3. Aktualisieren Sie die CNF-Zuordnungen:

   ```
   SETROPTS RACLIST(DIGTNMAP) REFRESH
   ```

Anmerkung:

1. Wenn das tatsächliche Zertifikat in derRACF-Datenbank gespeichert ist, wird die Benutzer-ID, unter der sie installiert ist, anstelle der einem Zertifikatsnamensfilter zugeordneten Benutzer-ID verwendet. Wenn das Zertifikat nicht in der RACF-Datenbank gespeichert ist, wird die Benutzer-ID verwendet, die dem zutreffendsten Zertifikatsnamensfilter entspricht. Übereinstimmungen mit dem registrierten Namen des Subjekten werden genauer als Übereinstimmung mit dem registrierten Namen des registrierten Benutzers betrachtet.

2. Änderungen an den CNF-Dateien gelten erst, wenn Sie die CNF-Zuordnungen aktualisiert haben.
3. Ein definierter Name stimmt mit dem DN-Filter in einem CNF nur überein, wenn der DN-Filter identisch mit dem *niedrigstwertigen Teil* des DN ist. Der niedrigstwertigste Teil des DN enthält die Attribute, die normalerweise am Ende des definierten Namens (DN) aufgelistet sind, die jedoch am Anfang des Zertifikats stehen.

   Beispiel: Der SDNFILTER `'O=IBM.C=UK'` ist in Betracht zu ziehen. Ein Subjekt-DN von `'CN=QM1.O=IBM.C=UK'` stimmt mit diesem Filter überein, aber ein Subjekt-DN von `'CN=QM1.O=IBM.L=Hursley.C=UK'` stimmt mit diesem Filter nicht überein.

   Der am wenigsten signifikante Teil einiger Zertifikate kann Felder enthalten, die nicht mit dem DN-Filter übereinstimmen. Sie können diese Zertifikate ausschließen, indem Sie im SSLPEER-Muster im Befehl DEFINE CHANNEL ein DN-Muster angeben.
4. Wenn die am zutreffendsten übereinstimmende Zertifikatsnamensliste für RACF als NOTRUST definiert ist, verwendet die Entität die Benutzer-ID, unter der der Kanalinitiator ausgeführt wird.
5. RACF verwendet das Zeichen `'.'` als Trennzeichen. IBM MQ verwendet ein Komma oder ein Semikolon.

Sie können CNFs definieren, um sicherzustellen, dass die Entität nie die Kanalbenutzer-ID auf den Standardwert setzt. Dies ist die Benutzer-ID, unter der der Kanalinitiator ausgeführt wird. Definieren Sie für jedes CA-Zertifikat in dem Schlüsselring, der der Entität zugeordnet ist, einen CNF mit einem IDNFILTER, der genau mit dem registrierten Namen des Zertifikats übereinstimmt, das von der Zertifizierungsstelle ausgestellt wurde. Dadurch wird sichergestellt, dass alle Zertifikate, die die Entität verwenden kann, mit mindestens einem dieser CNFs übereinstimmen. Dies liegt daran, dass alle diese Zertifikate entweder mit dem Schlüsselring verbunden sein müssen, der der Entität zugeordnet ist, oder von einer Zertifizierungsstelle ausgegeben werden muss, für die ein Zertifikat mit dem Schlüsselring verbunden ist, der der Entität zugeordnet ist.

Weitere Informationen zu den Befehlen, die Sie zum Bearbeiten von Zertifikatsnamensfiltern verwenden, finden Sie im Handbuch *SecureWay Security Server RACF Security Administrator's Guide*.

z/OS *Senderkanal und Übertragungswarteschlange auf QMA unter z/OS definieren*

Verwenden Sie die Befehle **DEFINE CHANNEL** und **DEFINE QLOCAL** , um die erforderlichen Objekte zu definieren.

Vorgehensweise

Geben Sie in QMA Befehle wie das folgende Beispiel aus:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Sender channel using TLS from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

Ergebnisse

Es werden ein Senderkanal, TO.WSMB und eine Übertragungswarteschlange (QMB) erstellt.

z/OS *Empfängerkanal auf QMB unter z/OS definieren*

Verwenden Sie den Befehl **DEFINE CHANNEL** , um das erforderliche Objekt einzurichten.

Vorgehensweise

Geben Sie auf WSMB einen Befehl wie im folgenden Beispiel aus:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS to QMB')
```

Ergebnisse

Es wird ein Empfängerkanal (TO.QMB) erstellt.

z/OS *Senderkanal auf QMA unter z/OS starten*

Falls erforderlich, starten Sie ein Empfangsprogramm und aktualisieren Sie die Sicherheit. Starten Sie dann den Kanal mit dem Befehl **START CHANNEL** .

Vorgehensweise

1. Optional: Wenn dies noch nicht geschehen ist, starten Sie ein Empfangsprogramm auf WSMB.

   Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Empfängerkanal, wenn er benötigt wird. Weitere Informationen zum Starten eines Listeners finden Sie im Abschnitt Kanallistener starten .
2. Optional: Wenn SSL/TLS-Kanäle bereits ausgeführt wurden, setzen Sie den Befehl `REFRESH SECURITY TYPE(SSL)` ab.

   Dadurch wird sichergestellt, dass alle Änderungen, die am Schlüsselrepository vorgenommen wurden, verfügbar sind.
3. Starten Sie den Kanal auf QMA mit dem Befehl `START CHANNEL(TO.QMB)`.

Ergebnisse

Der Senderkanal wird gestartet.

z/OS *Selbst signierte Zertifikat unter z/OS austauschen*

Tauschen Sie die Zertifikate aus, die Sie zuvor extrahiert Wenn Sie FTP verwenden, verwenden Sie das richtige Format.

Vorgehensweise

Übertragen Sie den CA-Teil des Zertifikats QM1 an das System QM2 und umgekehrt, z. B. per FTP.

Wenn Sie die Zertifikate mit FTP übertragen, müssen Sie dies im richtigen Format tun.

Übertragen Sie die folgenden Zertifikatstypen im *binären* Format:

- DER-codiertes binäres X.509
- PKCS #7 (CA-Zertifikate)
- PKCS #12 (persönliche Zertifikate)

Übertragen Sie die folgenden Zertifikatstypen im ASCII-Format:

- PEM (Privacy-Enhanced Mail)
- Base64-codierte X.509

z/OS *Senderkanal und Übertragungswarteschlange auf QM1 unter z/OS definieren*

Verwenden Sie die Befehle **DEFINE CHANNEL** und **DEFINE QLOCAL** , um die erforderlichen Objekte zu definieren.

Vorgehensweise

Geben Sie auf WSM1 Befehle wie das folgende Beispiel aus:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')

DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

Die CipherSpecs an jedem Ende des Kanals müssen identisch sein.

Nur der Parameter SSLCIPH ist obligatorisch, wenn der Kanal TLS verwenden soll. Informationen zu den zulässigen Werten für den Parameter SSLCIPH finden Sie im Abschnitt „CipherSpecs und CipherSuites in IBM MQ" auf Seite 44.

Ergebnisse

Es werden ein Senderkanal, QM1.TO.QM2 und eine Übertragungswarteschlange, QM2, erstellt.

z/OS *Empfängerkanal auf QM2 unter z/OS definieren*

Verwenden Sie den Befehl **DEFINE CHANNEL** , um das erforderliche Objekt einzurichten.

Vorgehensweise

Geben Sie auf QM2 einen Befehl wie im folgenden Beispiel ein:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from QM1 to QM2')
```

Der Name des Kanals muss dem Namen des im Abschnitt „Senderkanal und Übertragungswarteschlange auf QM1 unter z/OS definieren" auf Seite 351 definierten Senderkanals entsprechen und es muss dieselbe CipherSpec verwendet werden.

z/OS *Senderkanal auf QM1 unter z/OS starten*

Falls erforderlich, starten Sie ein Empfangsprogramm und aktualisieren Sie die Sicherheit. Starten Sie dann den Kanal mit dem Befehl **START CHANNEL** .

Vorgehensweise

1. Optional: Wenn Sie dies noch nicht getan haben, starten Sie ein Empfangsprogramm auf QM2.

   Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Empfängerkanal, wenn er benötigt wird. Informationen zum Starten eines Listeners finden Sie unter Kanallistener starten
2. Optional: Wenn SSL/TLS-Kanäle bereits ausgeführt wurden, setzen Sie den Befehl REFRESH SECURITY TYPE (SSL) ab.

   Dadurch wird sichergestellt, dass alle Änderungen, die am Schlüsselrepository vorgenommen wurden, verfügbar sind.
3. Starten Sie auf WSM1 den Kanal mit dem Befehl `START CHANNEL(QM1.TO.QM2)`.

Ergebnisse

Der Senderkanal wird gestartet.

z/OS *SSL- oder TLS-Umgebung unter z/OS aktualisieren*

Aktualisieren Sie die TLS-Umgebung auf WS-Manager QMA mit dem Befehl **REFRESH SECURITY** .

Vorgehensweise

Geben Sie auf WSMA den folgenden Befehl ein:

```
REFRESH SECURITY TYPE(SSL)
```

Dadurch wird sichergestellt, dass alle Änderungen, die am Schlüsselrepository vorgenommen wurden, verfügbar sind.

z/OS *Anonyme Verbindungen auf einem Empfängerkanal unter z/OS zulassen*

Verwenden Sie den Befehl **ALTER CHANNEL** , um die SSL-oder TLS-Clientauthentifizierung optional zu machen.

Vorgehensweise

Geben Sie auf WSMB den folgenden Befehl ein:

```
ALTER CHANNEL(TO.QMB) CHLTYPE(RCVR) SSLCAUTH(OPTIONAL)
```

z/OS *Senderkanal auf QM1 unter z/OS starten*

Falls erforderlich, starten Sie den Kanalinitiator, starten Sie ein Empfangsprogramm, und aktualisieren Sie die Sicherheit. Starten Sie dann den Kanal mit dem Befehl **START CHANNEL** .

Vorgehensweise

1. Optional: Wenn Sie dies noch nicht getan haben, starten Sie den Kanalinitiator.
2. Optional: Wenn Sie dies noch nicht getan haben, starten Sie ein Empfangsprogramm auf QM2.

   Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Empfängerkanal, wenn er benötigt wird. Informationen zum Starten eines Listeners finden Sie unter Kanallistener starten
3. Optional: Wenn der Kanalinitiator bereits aktiv war oder alle SSL/TLS-Kanäle bereits ausgeführt wurden, setzen Sie den Befehl REFRESH SECURITY TYPE (SSL) ab.

   Dadurch wird sichergestellt, dass alle Änderungen, die am Schlüsselrepository vorgenommen wurden, verfügbar sind.
4. Starten Sie auf WSM1 den Kanal mit dem Befehl `START CHANNEL(QM1.TO.QM2)`.

Ergebnisse

Der Senderkanal wird gestartet.

z/OS *Senderkanal auf QMA unter z/OS starten*

Falls erforderlich, starten Sie den Kanalinitiator, starten Sie ein Empfangsprogramm, und aktualisieren Sie die Sicherheit. Starten Sie dann den Kanal mit dem Befehl **START CHANNEL** .

Vorgehensweise

1. Optional: Starten Sie den Kanalinitiator, falls Sie dies noch nicht getan haben.
2. Optional: Wenn dies noch nicht geschehen ist, starten Sie ein Empfangsprogramm auf WSMB.

   Das Empfangsprogrammprogramm ist empfangsbereit für eingehende Netzanforderungen und startet den Empfängerkanal, wenn er benötigt wird. Weitere Informationen zum Starten eines Listeners finden Sie im Abschnitt Kanallistener starten .
3. Optional: Wenn der Kanalinitiator bereits aktiv war oder wenn SSL/TLS-Kanäle bereits ausgeführt wurden, setzen Sie den Befehl `REFRESH SECURITY TYPE(SSL)` ab.

   Dadurch wird sichergestellt, dass alle Änderungen, die am Schlüsselrepository vorgenommen wurden, verfügbar sind.
4. Starten Sie den Kanal auf QMA mit dem Befehl `START CHANNEL(TO.QMB)`.

Ergebnisse

Der Senderkanal wird gestartet.

z/OS *Schlüssellänge der elliptischen Kurve in z/OS ändern*

Wie Sie die Umgebungsvariable „GSK_CLIENT_ECURVE_LIST“ ändern, um die Liste der elliptischen Kurven oder unterstützten Gruppen, die vom Client angegeben werden, als eine Zeichenfolge festzulegen, die aus einem oder mehreren 4-Zeichen-Werten in der Reihenfolge der Benutzervorgabe zur Verwendung besteht.

Wichtig: Sie müssen den Fix in z/OS APAR OA61783 anwenden, damit bestimmte elliptische Kurven vom Betriebssystem wirksam werden, wenn TLS 1.0, TLS 1.1 und/oder TLS 1.2 vereinbarte Verbindungen verwendet werden.

Sie können diese TLS-Umgebungsvariable mithilfe der DD-Anweisung „CEEOPTS“ in der Start-JCL des Kanalinitiators festlegen:

```
CEEOPTS DD DSN=<dataset-name>,DISP=SHR
```

Geben Sie in dem oben angegebenen Dataset die Liste an, die Sie verwenden möchten, wie zum Beispiel:

```
ENVAR(“GSK_CLIENT_ECURVE_LIST=002300240025”)
```

Wichtig: Verwenden Sie diese Anweisung „CEEOPTS“ nicht bei In-Stream-Daten, da dadurch verhindert wird, dass die Umgebungsvariable für alle TLS-Tasks mit dieser Anweisung gesetzt wird.

Stellen Sie sicher, dass Sie ein sequenzielles Dataset oder Member einer partitionierten Datei referenzieren, damit dies bei Verwendung eines SSLTASKS-Werts größer als 1 funktioniert.

Sie können auch die serveranaloge Entsprechung von GSK_CLIENT_ECURVE_LIST verwenden, die GSK_SERVER_ALLOWED_KEX_ECURVES ist. Weitere Informationen finden Sie unter Elliptische Kurven für Schlüsselaustausch begrenzen .

Darüber hinaus finden Sie in Tabelle 5 im Abschnitt Cipher-Suite-Definitionen eine Liste der gültigen 4-stelligen elliptischen Kurven und unterstützten Gruppenspezifikationen.

Die Standardspezifikation lautet `00210023002400250019`. Wenn TLS Version 1.3 aktiviert ist, wird `0029 (x25519)` an das Ende der Standardliste angehängt.

Benutzer identifizieren und authentifizieren

Sie können Benutzer mithilfe von X.509-Zertifikaten, der MQCSP-Struktur oder in mehreren Typen von Benutzerexitprogrammen identifizieren und authentifizieren.

X.509-Zertifikate verwenden

Sie können Benutzer mithilfe von x.509 -Zertifikaten mit dem Befehl **CHLAUTH** und dem Parameter **SSLPEER** identifizieren und authentifizieren. Der Parameter **SSLPEER** gibt einen Filter an, der für den Vergleich mit dem definierten Namen des Zertifikats vom Peer-WS-Manager oder Client am anderen Ende des Kanals verwendet werden soll.

Weitere Informationen zur Verwendung des Befehls **CHLAUTH** und des Parameters **SSLPEER** finden Sie in SET CHLAUTH.

Verwenden der MQCSP-Struktur

Sie geben die Struktur der MQCSP-Verbindungssicherheitsparameter in einem MQCONNX-Aufruf an; diese Struktur enthält eine Benutzer-ID und ein Kennwort. Falls erforderlich, können Sie den MQCSP in einem Sicherheitsexit ändern.

Anmerkung: Der Objektberechtigungsmanager (Object Authority Manager, OAM) verwendet das Kennwort nicht. Der OAM funktioniert jedoch mit der Benutzer-ID, die als triviale Form der Authentifizierung betrachtet werden kann. Diese Prüfungen beenden die Übernahme einer anderen Benutzer-ID, wenn Sie diese Parameter in Ihren Anwendungen verwenden.

Warnung: In einigen Fällen wird das Kennwort in einer MQCSP-Struktur für eine Clientanwendung über ein Netz in Klartext gesendet. Um sicherzustellen, dass die Kennwörter der Clientanwendung ordnungsgemäß geschützt sind, finden Sie weitere Informationen in „MQCSP-Kennwortschutz” auf Seite 34.

Implementierung der Identifikation und Authentifizierung in Sicherheitsexits

Der primäre Zweck eines Sicherheitsexits besteht darin, den MCA an jedem Ende eines Kanals zu aktivieren, um seinen Partner zu authentifizieren. An jedem Ende eines Nachrichtenkanals und am Serverende eines MQI-Kanals handelt ein MCA in der Regel im Namen des Warteschlangenmanagers, mit dem er verbunden ist. Am Clientende eines MQI-Kanals handelt ein Nachrichtenkanalagent normalerweise im Namen des Benutzers der IBM MQ-Clientanwendung. Die gegenseitige Authentifizierung erfolgt in diesen Fällen zwischen zwei Warteschlangenmanagern oder zwischen einem Warteschlangenmanager und dem Benutzer einer IBM MQ MQI client-Anwendung.

Der angegebene Sicherheitsexit (der SSPI-Kanal-Exit) zeigt, wie die gegenseitige Authentifizierung implementiert werden kann, indem Authentifizierungstoken ausgetauscht werden, die von einem vertrauenswürdigen Authentifizierungsserver wie z. B. Kerberos generiert und anschließend überprüft werden. Weitere Informationen finden Sie unter „SSPI-Kanalexitprogramm unter Windows” auf Seite 166.

Die gegenseitige Authentifizierung kann auch mithilfe der PKI-Technologie (Public Key Infrastructure) implementiert werden. Jeder Sicherheitsexit generiert einige Zufallsdaten, signiert ihn mit dem privaten Schlüssel des Warteschlangenmanagers oder des Benutzers, der es darstellt, und sendet die signierten Daten an seinen Partner in einer Sicherheitsnachricht. Der Partner-Sicherheitsexit führt die Authentifizierung aus, indem er die digitale Signatur mit dem öffentlichen Schlüssel des Warteschlangenmanagers oder Benutzers überprüft. Vor dem Austausch von digitalen Signaturen müssen die Sicherheitsexits möglicherweise den Algorithmus für die Generierung eines Nachrichtenauszugs akzeptieren, wenn mehr als ein Algorithmus für die Verwendung verfügbar ist.

Wenn ein Sicherheitsexit die signierten Daten an seinen Partner sendet, muss er auch einige Möglichkeiten zum Identifizieren des Warteschlangenmanagers oder des Benutzers, der er darstellt, senden. Dies kann ein Distinguished Name oder sogar ein digitales Zertifikat sein. Wenn ein digitales Zertifikat gesendet wird, kann der Partner-Sicherheitsexit das Zertifikat überprüfen, indem er die Zertifikatskette mit dem Root-CA-Zertifikat arbeitet. Dadurch wird das Eigentumsrecht an dem öffentlichen Schlüssel, der zur Überprüfung der digitalen Signatur verwendet wird, gewährleistet.

Der Partner-Sicherheitsexit kann ein digitales Zertifikat nur prüfen, wenn es Zugriff auf ein Schlüsselrepository hat, das die verbleibenden Zertifikate in der Zertifikatskette enthält. Wenn kein digitales Zertifikat für den Warteschlangenmanager oder den Benutzer gesendet wird, muss ein digitales Zertifikat in dem Schlüsselrepository verfügbar sein, auf das der Sicherheitsexit der Partnerberechtigung zugreifen kann. Der Partner-Sicherheitsexit kann die digitale Signatur nicht überprüfen, es sei denn, er kann den öffentlichen Schlüssel des Unterzeichners finden.

Transport Layer Security (TLS) verwendet PKI-Techniken wie die eben beschriebenen. Weitere Informationen dazu, wie TLS eine Authentifizierung durchführt, finden Sie in „Konzepte der Transport Layer Security (TLS)” auf Seite 17.

Wenn ein vertrauenswürdiger Authentifizierungsserver oder eine PKI-Unterstützung nicht verfügbar ist, können andere Verfahren verwendet werden. Eine allgemeine Technik, die in Sicherheitsexits implementiert werden kann, verwendet einen symmetrischen Schlüsselalgorithmus.

Einer der Sicherheitsexits, Exit A, generiert eine Zufallszahl und sendet sie in einer Sicherheitsnachricht an seinen Partner-Sicherheitsexit, Exit B. Exit B verschlüsselt die Nummer mit Hilfe der Kopie eines Schlüssels, der nur den beiden Sicherheitsexits bekannt ist. Exit B sendet die verschlüsselte Nummer, um die Nachricht A in einer Sicherheitsnachricht mit einer zweiten Zufallszahl zu beenden, die Exit B generiert hat. Exit A prüft, ob die erste Zufallszahl korrekt verschlüsselt wurde, verschlüsselt die zweite Zufallszahl unter Verwendung ihrer Kopie des Schlüssels und sendet die verschlüsselte Zahl, um die Nachricht B in einer Sicherheitsnachricht zu beenden. Der Exit B prüft dann, ob die zweite Zufallszahl korrekt verschlüsselt wurde. Wenn ein Sicherheitsexit während dieses Austauschs nicht mit der Authentizität eines anderen verlassen wird, kann er den MCA anweisen, den Kanal zu schließen.

Ein Vorteil dieses Verfahrens besteht darin, dass während des Austausches kein Schlüssel oder Kennwort über die Kommunikationsverbindung gesendet wird. Ein Nachteil ist, dass es keine Lösung für das Prob-

lem gibt, wie der gemeinsam genutzte Schlüssel auf sichere Weise verteilt werden kann. Eine Lösung für dieses Problem wird in „Vertraulichkeit in Benutzerexitprogrammen implementieren” auf Seite 499 beschrieben. Eine ähnliche Technik wird in SNA für die gegenseitige Authentifizierung von zwei LUs verwendet, wenn sie eine Sitzung binden. Das Verfahren wird in „Authentifizierung auf Sitzungsebene” auf Seite 130 beschrieben.

Alle vorhergehenden Verfahren für die gegenseitige Authentifizierung können so angepasst werden, dass eine Einwegauthentifizierung möglich ist.

Identifikation und Authentifizierung in Nachrichtenexits implementieren

Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, enthält das Feld *UserIdentifier* im Nachrichtendeskriptor eine Benutzer-ID, die der Anwendung zugeordnet ist. Es sind jedoch keine Daten vorhanden, die zur Authentifizierung der Benutzer-ID verwendet werden können. Diese Daten können von einem Nachrichtenexit am sendenden Ende eines Kanals hinzugefügt und von einem Nachrichtenexit auf der Empfangsseite des Kanals überprüft werden. Die authentifizierenden Daten können beispielsweise ein verschlüsseltes Kennwort oder eine digitale Signatur sein.

Dieser Service ist möglicherweise effektiver, wenn er auf Anwendungsebene implementiert wird. Die grundlegende Voraussetzung ist, dass der Benutzer der Anwendung, der die Nachricht empfängt, den Benutzer der Anwendung, die die Nachricht gesendet hat, identifizieren und authentifizieren kann. Es ist daher selbstverständlich, die Umsetzung dieses Dienstes auf Anwendungsebene in Betracht zu ziehen. Weitere Informationen finden Sie unter „Identitätsabgleich im API-Exit und API-Steuerübergabeexit” auf Seite 361.

Implementierung der Identifikation und Authentifizierung in API-Exit und API-Steuerübergabeexit

Auf der Ebene einer einzelnen Nachricht ist die Identifikation und Authentifizierung ein Service, der zwei Benutzer, den Absender und den Empfänger der Nachricht umfasst. Die grundlegende Voraussetzung ist, dass der Benutzer der Anwendung, der die Nachricht empfängt, den Benutzer der Anwendung, die die Nachricht gesendet hat, identifizieren und authentifizieren kann. Beachten Sie, dass die Anforderung auf eine Art und Weise nicht auf zwei Weise authentifiziert wird.

Je nachdem, wie die Implementierung durchgeführt wird, müssen die Benutzer und ihre Anwendungen mit dem Service möglicherweise eine Schnittstelle oder sogar eine Interaktion mit dem Service benötigen. Darüber hinaus kann, wann und wie der Service verwendet wird, davon abhängen, wo sich die Benutzer und ihre Anwendungen befinden, sowie über die Art der Anwendungen selbst. Es ist daher selbstverständlich, die Implementierung des Service auf Anwendungsebene und nicht auf der Linkebene in Erwägung zu ziehen.

Wenn Sie die Implementierung dieses Service auf der Linkebene in Betracht ziehen, müssen Sie möglicherweise Probleme wie die folgenden beheben:

- Wie wenden Sie den Service in einem Nachrichtenkanal nur auf die Nachrichten an, die ihn benötigen?
- Wie können Benutzer und ihre Anwendungen mit dem Service eine Schnittstelle oder Interaktion mit dem Service aktivieren, wenn dies eine Voraussetzung ist?
- In einer Multi-Hop-Situation, in der eine Nachricht über mehr als einen Nachrichtenkanal auf dem Weg zum Ziel gesendet wird, wo rufen Sie die Komponenten des Service auf?

Im Folgenden finden Sie einige Beispiele dafür, wie der Identifizierungs-und Authentifizierungsservice auf Anwendungsebene implementiert werden kann. Der Begriff *API-Exit* bedeutet, dass entweder ein API-Exit oder ein API-Steuerübergabeexit vorhanden ist.

- Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, kann ein API-Exit ein Authentifizierungstoken von einem vertrauenswürdigen Authentifizierungsserver wie z. B. Kerberos anfordern. Der API-Exit kann dieses Token zu den Anwendungsdaten in der Nachricht hinzufügen. Wenn die Nachricht von der empfangenden Anwendung abgerufen wird, kann ein zweiter API-Exit den Authentifizierungsserver auffordern, den Sender zu authentifizieren, indem er das Token überprüft.

- Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, kann ein API-Exit die folgenden Elemente an die Anwendungsdaten in der Nachricht anhängen:
  - Das digitale Zertifikat des Absenders
  - Die digitale Signatur des Absenders

  Wenn verschiedene Algorithmen für die Generierung eines Nachrichten-Digest für die Verwendung verfügbar sind, kann der API-Exit den Namen des verwendeten Algorithmus enthalten.

  Wenn die Nachricht von der empfangenden Anwendung abgerufen wird, kann ein zweiter API-Exit die folgenden Prüfungen ausführen:
  - Der API-Exit kann das digitale Zertifikat überprüfen, indem er die Zertifikatskette mit dem Root-CA-Zertifikat arbeitet. Zu diesem Vorgang muss der API-Exit Zugriff auf ein Schlüsselrepository haben, das die verbleibenden Zertifikate in der Zertifikatskette enthält. Mit dieser Prüfung wird sichergestellt, dass der Absender, der durch den definierten Namen (Distinguished Name) identifiziert wird, der tatsächliche Eigner des öffentlichen Schlüssels ist, der im Zertifikat enthalten ist.
  - Der API-Exit kann die digitale Signatur mit Hilfe des öffentlichen Schlüssels überprüfen, der im Zertifikat enthalten ist. Bei dieser Prüfung wird der Absender authentifiziert.

  Der Distinguished Name des Absenders kann an Stelle des gesamten digitalen Zertifikats gesendet werden. In diesem Fall muss das Schlüsselrepository das Absenderzertifikat enthalten, damit der zweite API-Exit den öffentlichen Schlüssel des Absenders finden kann. Eine andere Möglichkeit besteht darin, alle Zertifikate in der Zertifikatskette zu senden.
- Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, enthält das Feld *UserIdentifier* im Nachrichtendeskriptor eine Benutzer-ID, die der Anwendung zugeordnet ist. Die Benutzer-ID kann zum Identifizieren des Absenders verwendet werden. Um die Authentifizierung zu aktivieren, kann ein API-Exit einige Daten, wie z. B. ein verschlüsseltes Kennwort, an die Anwendungsdaten in der Nachricht anhängen. Wenn die Nachricht von der empfangenden Anwendung abgerufen wird, kann ein zweiter API-Exit die Benutzer-ID authentifizieren, indem die Daten verwendet werden, die mit der Nachricht gereist sind.

  Diese Technik kann als ausreichend für Nachrichten betrachtet werden, die aus einer kontrollierten und vertrauenswürdigen Umgebung stammen, und in Fällen, in denen ein anerkannter Authentifizierungsserver oder PKI-Unterstützung nicht verfügbar ist.

Pluggable Authentication Method (PAM)

Linux AIX

PAM ist jetzt auf allen UNIX and Linux-Plattformen vorhanden und stellt ein allgemeines Verfahren bereit, mit dem die Details der Benutzerauthentifizierung aus Services ausgeblendet werden können.

Verschiedene Authentifizierungsregeln können für verschiedene Services verwendet werden, indem die Regeln konfiguriert werden, ohne dass Änderungen an den Services selbst erforderlich sind.

Im Abschnitt „Verwenden der Pluggable Authentication Method (PAM)” auf Seite 375 finden Sie weitere Informationen.

Privilegierte Benutzer

Ein privilegierter Benutzer hat vollständige Administratorberechtigungen für IBM MQ.

Zusätzlich zu den in der folgenden Tabelle aufgeführten Benutzern gibt es bestimmte Objekte und Berechtigungen, für die beim Erteilen von Zugriffsberechtigungen zusätzliche Sorgfalt erstellt werden muss, um die Integrität und Sicherheit des Warteschlangenmanagers zu gewährleisten. Eine zusätzliche Überprüfung ist erforderlich, wenn Sie eine der folgenden Berechtigungen erteilen:

- Alle Berechtigungen für SYSTEM -Objekte
- Verwaltungsberechtigungen zum Erstellen, Ändern und Löschen von Objekten.

z/OS Unter z/OS hat diese Berechtigung die Befehlssicherheit und die Sicherheit der Befehlsressourcen, um DEFINE-, ALTER-und DELETE-Befehle auszugeben.

Multi Auf allen anderen Plattformen sind diese Berechtigungen Verwaltungsberechtigungen, wie z. B. `+crt`, `+chg` und `+dlt`.

- Verwaltungsberechtigung zum Löschen von Warteschlangen.

  z/OS Unter z/OS ist diese Berechtigung die Autorität der Befehlssicherheit und Sicherheit der Befehlsressourcen, um CLEAR-Befehle auszugeben.

  Multi Auf allen anderen Plattformen ist diese Berechtigung `+clr`.
- Verwaltungsberechtigungen zum Stoppen von Kanälen, Zurückschreibungsnachrichten oder Festschreiben von Nachrichten.

  z/OS Unter z/OS ist diese Berechtigung die Autorität für die Befehlssicherheit und Sicherheit der Befehlsressourcen, um Befehle wie RESET CHANNEL, START CHANNEL und STOP CHANNEL auszugeben.

  Multi Auf allen anderen Plattformen sind diese Berechtigungen `+ctrl` und `+ctrlx`.
- Alternative Benutzer-MQI-Berechtigung, die es Anwendungen ermöglicht, Berechtigungen für Berechtigungsprüfungen zu eskalieren.

  z/OS Unter z/OS ist diese Berechtigung eine Autorität, die den alternativen Benutzersicherheitsprofilen erteilt wird.

  Multi Auf allen anderen Plattformen ist diese Berechtigung `+altusr`.
- Kontextberechtigungen, die es Anwendungen ermöglichen, den Sicherheitskontext von Nachrichten zu ändern.

  z/OS Unter z/OS ist diese Berechtigung eine Autorität, die den Kontextsicherheitsprofilen erteilt wird.

  Multi Auf allen anderen Plattformen sind diese Berechtigungen `+setall` und `+setid`.

Als allgemeiner Principal sollten Messaging-Anwendungen nur die grundlegenden MQI-Berechtigungen für die Warteschlangen oder Themen erhalten, die benötigt werden. MCA-Kanäle, die unter einem nicht privilegierten MCAUSER ausgeführt werden, und bestimmte andere spezielle Typen von Anwendungen, wie z. B. für Warteschlangen-Handler, erfordern möglicherweise zusätzliche Berechtigungen, die normalerweise nicht für Anwendungen erteilt werden, um ordnungsgemäß zu funktionieren.

Tabelle 67. Privilegierte Benutzer nach Plattform

Plattform	**Privilegierte Benutzer**
Systeme mit Windows	• SYSTEM • Mitglieder der Gruppe 'mqm' • Mitglieder der Gruppe Administratoren
Systeme mit AIX and Linux	• Mitglieder der Gruppe 'mqm'
IBM i IBM i Systeme mit IBM i	• Die Profile qmqm und qmqmadm • Alle Mitglieder der Gruppe 'qmqmadm' • Jeder Benutzer, der mit der Einstellung *ALLOBJ definiert wurde

Tabelle 67. Privilegierte Benutzer nach Plattform (Forts.)	
Plattform	**Privilegierte Benutzer**
z/OS	Die Benutzer-ID, unter der der Kanalinitiator, der WS-Manager und die Sicherheitsadressräume für erweiterte Nachrichten ausgeführt werden. Diese Benutzer-IDs verfügen nicht automatisch über vollständige Administratorberechtigungen für IBM MQ, sondern werden aufgrund der Zugriffsebene, die diesen Benutzer-IDs in der Regel erteilt wird, als privilegiert betrachtet.

Benutzer mit der MQCSP-Struktur identifizieren und authentifizieren

Sie können die Struktur der MQCSP-Verbindungssicherheitsparameter in einem MQCONNX-Aufruf angeben.

Die Struktur der MQCSP-Verbindungssicherheitsparameter enthält eine Benutzer-ID und ein Kennwort, die der Berechtigungsservice zur Identifizierung und Authentifizierung des Benutzers verwenden kann.

Sie können den MQCSP in einem Sicherheitsexit ändern.

Warnung: In einigen Fällen wird das Kennwort in einer MQCSP-Struktur für eine Clientanwendung über ein Netz in Klartext gesendet. Um sicherzustellen, dass die Kennwörter der Clientanwendung ordnungsgemäß geschützt sind, finden Sie weitere Informationen in „MQCSP-Kennwortschutz” auf Seite 34.

Beziehung zwischen MQCSP- und AdoptCTX-Einstellungen

IBM MQ authentifiziert immer Berechtigungsnachweise, die über die MQCSP-Struktur übergeben werden, es sei denn, die Verbindungsauthentifizierungsfunktion ist nicht aktiviert. Nachdem die Berechtigungsnachweise erfolgreich authentifiziert wurden, versucht IBM MQ, die Benutzer-ID für künftige Berechtigungsprüfungen zu übernehmen, es sei denn, ADOPTCTX ist nicht aktiviert.

IBM MQ hat einen Grenzwert für die Länge der Benutzer-IDs, die Benutzer für Berechtigungsprüfungen verwenden können. Diese Grenzwerte sind in „Benutzer-IDs” auf Seite 93 detailliert beschrieben. Bei der Übernahme einer Benutzer-ID, die durch die MQCSP-Struktur gegangen ist, verhält sich IBM MQ anders und in Abhängigkeit der anderen Konfigurationsoptionen:

- Bei Verwendung der LDAP-Verbindungsauthentifizierung ruft IBM MQ den Wert des in SHORTUSR festgelegten Felds aus dem LDAP-Datensatz des Benutzers für diesen Benutzer ab und übernimmt diese Benutzer-ID.

  Wenn SHORTUSR beispielsweise auf `'CN'` gesetzt ist und ein LDAP-Datensatz einen Benutzer als `'CN=Test,SN=MQ,O=IBM,C=UK'` auflistet, wird die Benutzer-ID `Test` verwendet.
- Wenn bei Verwendung der Verbindungsauthentifizierung des Betriebssystems oder der PAM-Authentifizierung ADOPTCTX auf YES gesetzt ist, wird die Benutzer-ID, die über die MQCSP-Struktur übergeben wird, abgeschnitten, damit die 12 Zeichen lange Benutzer-ID-Begrenzung von IBM MQ eingehalten wird, wenn sie als Verbindungskontext übernommen wird.

  Wenn **ChlAuthEarlyAdopt** aktiviert ist, erfolgt das Abschneiden nach der Authentifizierung der Benutzerberechtigungsnachweise.

  Wenn **ChlAuthEarlyAdopt** nicht aktiviert ist, erfolgt das Abschneiden vor der Übernahme. Wenn der Benutzer unter Windowsim Format `user@domain`angegeben wird, bedeutet dies, dass das Abschneiden zu einer Domänenspezifikation führen kann, die nicht gültig ist, wenn der Benutzer weniger als 12 Zeichen hat.

  Wenn beispielsweise der Benutzer `` `ibmmq@windowsdomain` `` über den MQCSP bereitgestellt wird, wird er in diesem Szenario auf `` `ibmmq@window` `` abgeschnitten. Dies führt zu folgendem Fehler:

  ```
  AMQ8074W: Die Berechtigung ist fehlgeschlagen, da die SID 'SID' nicht mit der Entität
  'ibmmq@window' übereinstimmt.
  ```

Wenn Sie auf dieser Basis eine Benutzer-ID mit mehr als 12 Zeichen (z. B. eine Windows -Domänenbenutzer-ID im Format `user@domain`) über den MQCSP übergeben, sollten Sie **ChlAuthEarlyAdopt**=*Y* in der Datei qm.ini konfigurieren, um diesen Fehler zu vermeiden.

Alternativ können Sie ADOPTCTX (NO) in der CONNAUTH AUTHINFO-Konfiguration verwenden und eine alternative Methode wie eine CHLAUTH USERMAP-Regel, einen Sicherheitsexit oder die Einstellung des Kanalobjekts MCAUSER verwenden, um die Benutzer-ID für den Kanal festzulegen.

Implementierung der Identifikation und Authentifizierung in Sicherheitsexits

Sie können einen Sicherheitsexit verwenden, um eine Einweg-oder gegenseitige Authentifizierung zu implementieren.

Der primäre Zweck eines Sicherheitsexits besteht darin, den MCA an jedem Ende eines Kanals zu aktivieren, um seinen Partner zu authentifizieren. An jedem Ende eines Nachrichtenkanals und am Serverende eines MQI-Kanals handelt ein MCA in der Regel im Namen des Warteschlangenmanagers, mit dem er verbunden ist. Am Clientende eines MQI-Kanals handelt ein Nachrichtenkanalagent normalerweise im Namen des Benutzers der IBM MQ MQI client-Anwendung. Die gegenseitige Authentifizierung erfolgt in diesen Fällen zwischen zwei Warteschlangenmanagern oder zwischen einem Warteschlangenmanager und dem Benutzer einer IBM MQ MQI client-Anwendung.

Der angegebene Sicherheitsexit (der SSPI-Kanal-Exit) zeigt, wie die gegenseitige Authentifizierung implementiert werden kann, indem Authentifizierungstoken ausgetauscht werden, die von einem vertrauenswürdigen Authentifizierungsserver wie z. B. Kerberos generiert und anschließend überprüft werden. Weitere Informationen finden Sie unter „SSPI-Kanalexitprogramm unter Windows” auf Seite 166.

Die gegenseitige Authentifizierung kann auch mithilfe der PKI-Technologie (Public Key Infrastructure) implementiert werden. Jeder Sicherheitsexit generiert einige Zufallsdaten, signiert ihn mit dem privaten Schlüssel des Warteschlangenmanagers oder des Benutzers, der es darstellt, und sendet die signierten Daten an seinen Partner in einer Sicherheitsnachricht. Der Partner-Sicherheitsexit führt die Authentifizierung aus, indem er die digitale Signatur mit dem öffentlichen Schlüssel des Warteschlangenmanagers oder Benutzers überprüft. Vor dem Austausch von digitalen Signaturen müssen die Sicherheitsexits möglicherweise den Algorithmus für die Generierung eines Nachrichtenauszugs akzeptieren, wenn mehr als ein Algorithmus für die Verwendung verfügbar ist.

Wenn ein Sicherheitsexit die signierten Daten an seinen Partner sendet, muss er auch einige Möglichkeiten zum Identifizieren des Warteschlangenmanagers oder des Benutzers, der er darstellt, senden. Dies kann ein Distinguished Name oder sogar ein digitales Zertifikat sein. Wenn ein digitales Zertifikat gesendet wird, kann der Partner-Sicherheitsexit das Zertifikat überprüfen, indem er die Zertifikatskette mit dem Root-CA-Zertifikat arbeitet. Dadurch wird das Eigentumsrecht an dem öffentlichen Schlüssel, der zur Überprüfung der digitalen Signatur verwendet wird, gewährleistet.

Der Partner-Sicherheitsexit kann ein digitales Zertifikat nur prüfen, wenn es Zugriff auf ein Schlüsselrepository hat, das die verbleibenden Zertifikate in der Zertifikatskette enthält. Wenn kein digitales Zertifikat für den Warteschlangenmanager oder den Benutzer gesendet wird, muss ein digitales Zertifikat in dem Schlüsselrepository verfügbar sein, auf das der Sicherheitsexit der Partnerberechtigung zugreifen kann. Der Partner-Sicherheitsexit kann die digitale Signatur nicht überprüfen, es sei denn, er kann den öffentlichen Schlüssel des Unterzeichners finden.

Transport Layer Security (TLS) verwendet PKI-Techniken wie die eben beschriebenen. Weitere Informationen zur Authentifizierung von Secure Sockets Layer finden Sie in „Konzepte der Transport Layer Security (TLS)” auf Seite 17.

Wenn ein vertrauenswürdiger Authentifizierungsserver oder eine PKI-Unterstützung nicht verfügbar ist, können andere Verfahren verwendet werden. Eine allgemeine Technik, die in Sicherheitsexits implementiert werden kann, verwendet einen symmetrischen Schlüsselalgorithmus.

Einer der Sicherheitsexits, Exit A, generiert eine Zufallszahl und sendet sie in einer Sicherheitsnachricht an seinen Partner-Sicherheitsexit, Exit B. Exit B verschlüsselt die Nummer mit Hilfe der Kopie eines Schlüssels, der nur den beiden Sicherheitsexits bekannt ist. Exit B sendet die verschlüsselte Nummer, um die Nachricht A in einer Sicherheitsnachricht mit einer zweiten Zufallszahl zu beenden, die Exit B generiert

hat. Exit A prüft, ob die erste Zufallszahl korrekt verschlüsselt wurde, verschlüsselt die zweite Zufallszahl unter Verwendung ihrer Kopie des Schlüssels und sendet die verschlüsselte Zahl, um die Nachricht B in einer Sicherheitsnachricht zu beenden. Der Exit B prüft dann, ob die zweite Zufallszahl korrekt verschlüsselt wurde. Wenn ein Sicherheitsexit während dieses Austauschs nicht mit der Authentizität eines anderen verlassen wird, kann er den MCA anweisen, den Kanal zu schließen.

Ein Vorteil dieses Verfahrens besteht darin, dass während des Austausches kein Schlüssel oder Kennwort über die Kommunikationsverbindung gesendet wird. Ein Nachteil ist, dass es keine Lösung für das Problem gibt, wie der gemeinsam genutzte Schlüssel auf sichere Weise verteilt werden kann. Eine Lösung für dieses Problem wird in „Vertraulichkeit in Benutzerexitprogrammen implementieren” auf Seite 499 beschrieben. Eine ähnliche Technik wird in SNA für die gegenseitige Authentifizierung von zwei LUs verwendet, wenn sie eine Sitzung binden. Das Verfahren wird in „Authentifizierung auf Sitzungsebene” auf Seite 130 beschrieben.

Alle vorhergehenden Verfahren für die gegenseitige Authentifizierung können so angepasst werden, dass eine Einwegauthentifizierung möglich ist.

Identitätsabgleich in Nachrichtenexits

Sie können Nachrichtenexits verwenden, um Informationen zu verarbeiten, um eine Benutzer-ID zu authentifizieren. Es kann jedoch besser sein, die Authentifizierung auf Anwendungsebene zu implementieren.

Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, enthält das Feld *UserIdentifier* im Nachrichtendeskriptor eine Benutzer-ID, die der Anwendung zugeordnet ist. Es sind jedoch keine Daten vorhanden, die zur Authentifizierung der Benutzer-ID verwendet werden können. Diese Daten können von einem Nachrichtenexit am sendenden Ende eines Kanals hinzugefügt und von einem Nachrichtenexit auf der Empfangsseite des Kanals überprüft werden. Die authentifizierenden Daten können beispielsweise ein verschlüsseltes Kennwort oder eine digitale Signatur sein.

Dieser Service ist möglicherweise effektiver, wenn er auf Anwendungsebene implementiert wird. Die grundlegende Voraussetzung ist, dass der Benutzer der Anwendung, der die Nachricht empfängt, den Benutzer der Anwendung, die die Nachricht gesendet hat, identifizieren und authentifizieren kann. Es ist daher selbstverständlich, die Umsetzung dieses Dienstes auf Anwendungsebene in Betracht zu ziehen. Weitere Informationen finden Sie in „Identitätsabgleich im API-Exit und API-Steuerübergabeexit” auf Seite 361.

Identitätsabgleich im API-Exit und API-Steuerübergabeexit

Eine Anwendung, die eine Nachricht empfängt, muss in der Lage sein, den Benutzer der Anwendung, die die Nachricht gesendet hat, zu identifizieren und zu authentifizieren. Dieser Service wird in der Regel am besten auf Anwendungsebene implementiert. API-Exits können den Service in einer Reihe von Methoden implementieren.

Auf der Ebene einer einzelnen Nachricht ist die Identifikation und Authentifizierung ein Service, der zwei Benutzer, den Absender und den Empfänger der Nachricht umfasst. Die grundlegende Voraussetzung ist, dass der Benutzer der Anwendung, der die Nachricht empfängt, den Benutzer der Anwendung, die die Nachricht gesendet hat, identifizieren und authentifizieren kann. Beachten Sie, dass die Anforderung auf eine Art und Weise nicht auf zwei Weise authentifiziert wird.

Je nachdem, wie die Implementierung durchgeführt wird, müssen die Benutzer und ihre Anwendungen mit dem Service möglicherweise eine Schnittstelle oder sogar eine Interaktion mit dem Service benötigen. Darüber hinaus kann, wann und wie der Service verwendet wird, davon abhängen, wo sich die Benutzer und ihre Anwendungen befinden, sowie über die Art der Anwendungen selbst. Es ist daher selbstverständlich, die Implementierung des Service auf Anwendungsebene und nicht auf der Linkebene in Erwägung zu ziehen.

Wenn Sie die Implementierung dieses Service auf der Linkebene in Betracht ziehen, müssen Sie möglicherweise Probleme wie die folgenden beheben:

- Wie wenden Sie den Service in einem Nachrichtenkanal nur auf die Nachrichten an, die ihn benötigen?

- Wie können Benutzer und ihre Anwendungen mit dem Service eine Schnittstelle oder Interaktion mit dem Service aktivieren, wenn dies eine Voraussetzung ist?
- In einer Multi-Hop-Situation, in der eine Nachricht über mehr als einen Nachrichtenkanal auf dem Weg zum Ziel gesendet wird, wo rufen Sie die Komponenten des Service auf?

Im Folgenden finden Sie einige Beispiele dafür, wie der Identifizierungs-und Authentifizierungsservice auf Anwendungsebene implementiert werden kann. Der Begriff *API-Exit* bedeutet, dass entweder ein API-Exit oder ein API-Steuerübergabeexit vorhanden ist.

- Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, kann ein API-Exit ein Authentifizierungstoken von einem vertrauenswürdigen Authentifizierungsserver wie z. B. Kerberos anfordern. Der API-Exit kann dieses Token zu den Anwendungsdaten in der Nachricht hinzufügen. Wenn die Nachricht von der empfangenden Anwendung abgerufen wird, kann ein zweiter API-Exit den Authentifizierungsserver auffordern, den Sender zu authentifizieren, indem er das Token überprüft.
- Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, kann ein API-Exit die folgenden Elemente an die Anwendungsdaten in der Nachricht anhängen:
  - Das digitale Zertifikat des Absenders
  - Die digitale Signatur des Absenders

  Wenn verschiedene Algorithmen für die Generierung eines Nachrichten-Digest für die Verwendung verfügbar sind, kann der API-Exit den Namen des verwendeten Algorithmus enthalten.

  Wenn die Nachricht von der empfangenden Anwendung abgerufen wird, kann ein zweiter API-Exit die folgenden Prüfungen ausführen:

  - Der API-Exit kann das digitale Zertifikat überprüfen, indem er die Zertifikatskette mit dem Root-CA-Zertifikat arbeitet. Zu diesem Vorgang muss der API-Exit Zugriff auf ein Schlüsselrepository haben, das die verbleibenden Zertifikate in der Zertifikatskette enthält. Mit dieser Prüfung wird sichergestellt, dass der Absender, der durch den definierten Namen (Distinguished Name) identifiziert wird, der tatsächliche Eigner des öffentlichen Schlüssels ist, der im Zertifikat enthalten ist.
  - Der API-Exit kann die digitale Signatur mit Hilfe des öffentlichen Schlüssels überprüfen, der im Zertifikat enthalten ist. Bei dieser Prüfung wird der Absender authentifiziert.

  Der Distinguished Name des Absenders kann an Stelle des gesamten digitalen Zertifikats gesendet werden. In diesem Fall muss das Schlüsselrepository das Absenderzertifikat enthalten, damit der zweite API-Exit den öffentlichen Schlüssel des Absenders finden kann. Eine andere Möglichkeit besteht darin, alle Zertifikate in der Zertifikatskette zu senden.
- Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, enthält das Feld *UserIdentifier* im Nachrichtendeskriptor eine Benutzer-ID, die der Anwendung zugeordnet ist. Die Benutzer-ID kann zum Identifizieren des Absenders verwendet werden. Um die Authentifizierung zu aktivieren, kann ein API-Exit einige Daten, wie z. B. ein verschlüsseltes Kennwort, an die Anwendungsdaten in der Nachricht anhängen. Wenn die Nachricht von der empfangenden Anwendung abgerufen wird, kann ein zweiter API-Exit die Benutzer-ID authentifizieren, indem die Daten verwendet werden, die mit der Nachricht gereist sind.

  Diese Technik kann als ausreichend für Nachrichten betrachtet werden, die aus einer kontrollierten und vertrauenswürdigen Umgebung stammen, und in Fällen, in denen ein anerkannter Authentifizierungsserver oder PKI-Unterstützung nicht verfügbar ist.

Mit widerrufenden Zertifikaten arbeiten

Digitale Zertifikate können von den Zertifizierungsstellen entzogen werden. Abhängig von der Plattform können Sie den Widerrufstatus von Zertifikaten mit OCSP oder CRLs auf LDAP-Servern überprüfen.

Während des TLS-Handshake authentifizieren sich die kommunizierenden Partner gegenseitig mit digitalen Zertifikaten. Die Authentifizierung kann eine Überprüfung enthalten, dass das empfangene Zertifikat immer noch vertrauenswürdig ist. Zertifizierungsstellen (CAs) entziehen die Zertifikate aus verschiedenen Gründen, z. B.:

- Der Eigner wurde in eine andere Organisation verschoben

- Der private Schlüssel ist nicht mehr geheim.

CAs veröffentlichen widerrufliche persönliche Zertifikate in einer Zertifikatswiderrufsliste (Certificate Revocation List, CRL). CA-Zertifikate, die widerrufen wurden, werden in einer Berechtigungswiderrufsliste (ARL, Authority Revocation List, Berechtigungswiderrufsliste) veröffentlicht.

ALW Auf AIX, Linux, and Windows-Plattformen überprüft die IBM MQ-SSL-Unterstützung mithilfe von OCSP (Online Certificate Status Protocol) oder mithilfe von CRLs und ARLs auf LDAP-Servern (Lightweight Directory Access Protocol), ob widerrufene Zertifikate vorhanden sind. OCSP ist die bevorzugte Methode.

IBM MQ classes for Java und IBM MQ classes for JMS können die OCSP-Informationen in einer Definitionstabelle für den Clientkanal nicht verwenden. Sie können OCSP jedoch wie im Abschnitt Online-Zertifikatprotokoll verwenden beschrieben konfigurieren.

z/OS IBM i Auf IBM i- und z/OS-Plattformen überprüft die IBM MQ-SSL-Unterstützung nur für CRLs und ARLs auf LDAP-Servern, ob widerrufene Zertifikate vorhanden sind.

Weitere Informationen zu Zertifizierungsstellen finden Sie in „Digitale Zertifikate” auf Seite 12.

OCSP/CRL-Prüfung

Die Überprüfung von Online Certificate Status Protocol (OCSP)/Certificate Revocation List (CRL) wird für ferne eingehende Zertifikate ausgeführt. Der Prozess überprüft die gesamte Kette vom persönlichen Zertifikat des fernen Systems bis zu seinem Stammzertifikat.

OCSP-Validierung mithilfe von openSSL prüfen

Wenn Ihr Unternehmen openSSL verwendet, um OCSP zu validieren, und Sie dann versuchen, eine GSKit-TLS-Verbindung zu verwenden, erhalten Sie die Statuswarnung „UNKNOWN“.

Dies liegt daran, dass alle Zertifikate in der Kette, abgesehen vom Stammelement, von GSKit auf den Widerrufsstatus überprüft werden. Die GSKit-Operation entspricht RFC 5280 – dies wird in der GSKit-Trust-Richtlinie beschrieben. Der GSKit-Algorithmus prüft alle verfügbaren Quellen auf Widerrufsinformationen, wie in RFC 5280 und der GSKit-Trust-Richtlinie beschrieben.

Wie funktioniert die OCSP/CRL-Prüfung in IBM MQ?

IBM MQ unterstützt zwei Mechanismen zur Steuerung des Verhaltens, wenn Zertifikate für benannte OCSP- oder CRL-Endpunkte überprüft werden, entweder in der Zertifikatserweiterung oder wie in den AUTHINFO-Objekten definiert:

- Die Attribute **OCSPCheckExtensions**, **CDPCheckExtensions**und **OCSPAuthentication** der Zeilengruppe SSL in der Datei qm.iniund
- Verwendung des Parameters SSLCRLNL des Warteschlangenmanagers und der Konfigurationen AUTHINFO OCSP und CRLLDAP. Weitere Informationen finden Sie unter ALTER AUTHINFO und ALTER QMGR .

Achtung:

Der ALTER AUTHINFO-Befehl mit **AUTHTYPE(OCSP)** gilt nicht für die Verwendung auf IBM i- oder z/OS-Queue Managern. Ein solches Objekt kann aber auch auf diesen Plattformen angegeben werden, um es für die Verwendung durch Clients in die Definitionstabelle für Clientkanäle (CCDT) zu kopieren.

Die SSL-Zeilengruppenattribute **OCSPCheckExtensions** und **CDPCheckExtensions** steuern, ob IBM MQ ein Zertifikat gegen den OCSP- oder CRL-Server überprüft, das in der AIA-Erweiterung des Zertifikats detailliert beschrieben wird.

Wenn diese Option nicht aktiviert ist, wird der OCSP- oder CRL-Server in der Zertifikatserweiterung nicht kontaktiert.

Wenn OCSP- oder CRL-Server über AUTHINFO-Objekte detailliert und mit dem Attribut "SSLCRLNL **QMGR**" referenziert werden, versucht IBM MQ während der Zertifikatswiderrufverarbeitung, diese Server zu kontaktieren.

Wichtig: Es kann nur ein OCSP-AUTHINFO-Objekt in der Namensliste SSLCRLNL definiert werden.

Wenn:

OCSPCheckExtensions=NO und **CDPCheckExtensions**=*NO* festgelegt sind und
Keine OCSP- oder CRL-Server in AUTHINFO-Objekten definiert sind

, dann wird keine Zertifikatswiderrufsprüfung durchgeführt.

Wenn Sie ein Zertifikat für seinen Widerrufsstatus überprüfen, kontaktiert IBM MQ die OCSP- oder CRL-Server, die in der folgenden Reihenfolge benannt sind, falls aktiviert:

1. Der OCSP-Server, der in einem **AUTHTYPE(OCSP)**-Objekt ausführlich beschrieben ist und im Attribut SSLCRLNL **QMGR** referenziert wird.
2. OCSP-Server, die in der AIA-Erweiterung der Zertifikats detailliert beschrieben werden, wenn **OCSPCheckExtensions**=*YES*.
3. CRL-Server, die in der **CRLDistributionPoints**-Erweiterung der Zertifikate detailliert beschrieben werden, wenn **CDPCheckExtensions** =*YES*.
4. Alle CRL-Server, die in **AUTHINFO(CRLLDAP)**-Objekten detailliert beschrieben und im Attribut SSLCRLNL **QMGR** referenziert werden.

Wenn bei der Verifizierung eines Zertifikats ein Schritt dazu führt, dass der OCSP-Server oder der CRL-Server eine endgültige REVOKED- oder VALID-Antwort auf eine Abfrage für das Zertifikat zurückgibt, werden keine weiteren Prüfungen durchgeführt und der Status des Zertifikats wird wie dargestellt verwendet, um festzustellen, ob das Zertifikat vertrauenswürdig ist oder nicht.

Wenn ein OCSP-Server oder CRL-Server ein Ergebnis UNKNOWN zurückgibt, wird die Verarbeitung fortgesetzt, bis ein OCSP- oder CRL-Server ein endgültiges Ergebnis zurückgibt oder alle Optionen erschöpft sind.

Das Verhalten, ob ein Zertifikat als widerrufen angesehen wird, wenn dessen Status nicht ermittelt werden kann, ist für OCSP- und CRL-Server unterschiedlich:

- Wenn für CRL-Server keine CRL abgerufen werden kann, wird das Zertifikat als NOT_REVOKED betrachtet.
- Wenn für OCSP-Server kein Widerrufstatus von einem benannten OCSP-Server abgerufen werden kann, wird das Verhalten über das Attribut **OCSPAuthentication** in der SSL-Zeilengruppe der Datei 'qm.ini' gesteuert.

  Sie können dieses Attribut so konfigurieren, dass es eine Verbindung blockiert, eine Verbindung zulässt oder eine Verbindung mit einer Warnung zulässt.

Sie können das Attribut **SSLHTTPProxyName**=*string* in der SSL-Zeilengruppe der Dateien 'qm.ini' und 'mqclient.ini' für die OCSP-Prüfungen verwenden, falls erforderlich. Bei der Zeichenfolge handelt es sich um den Hostnamen oder die Netzadresse des HTTP-Proxy-Servers, der von GSKit für OCSP-Prüfungen verwendet werden soll.

Ab IBM MQ 9.1.5 können Sie den Wert **OCSPTimeout** in der SSL-Zeilengruppe der Datei qm.ini oder mqclient.ini festlegen, die die Anzahl der Sekunden festlegen, die auf einen OCSP-Responder gewartet werden soll, wenn eine Widerrufsprüfung durchgeführt wird.

ALW Widerruftes Zertifikat und OCSP

IBM MQ ermittelt, welcher OCSP-Responder (Online Certificate Status Protocol) verwendet werden soll und verarbeitet die empfangene Antwort. Möglicherweise müssen Sie die Schritte ausführen, um den OCSP-Responder zugänglich zu machen.

Anmerkung: Diese Informationen gelten nur für IBM MQ auf Systemen mit AIX, Linux, and Windows.

Um den Widerrufsstatus eines digitalen Zertifikats mithilfe von OCSP zu überprüfen, kann IBM MQ zwei Methoden verwenden, mit denen bestimmt wird, welcher OCSP-Responder kontaktiert werden soll:

- Durch Verwendung der Zertifikatserweiterung "AuthorityInfoAccess (AIA)" in dem Zertifikat, das überprüft werden soll.
- Durch Verwendung einer URL, die in einem Authentifizierungsinformationsobjekt angegeben oder von einer Clientanwendung angegeben wird.

Eine URL, die in einem Authentifizierungsdatenobjekt oder von einer Clientanwendung angegeben wird, hat Vorrang vor einer URL in einer AIA-Zertifikatserweiterung.

Wenn die URL des OCSP-Responder hinter einer Firewall liegt, rekonfigurieren Sie die Firewall so, dass der OCSP-Responder auf einen OCSP-Proxy-Server zugreifen oder diese einrichten kann. Geben Sie den Namen des Proxy-Servers mithilfe der Variablen 'SSLHTTPProxyName' in der SSL-Zeilengruppe an. Auf Clientsystemen können Sie den Namen des Proxy-Servers auch mithilfe der Umgebungsvariablen MQSSLPROXY angeben. Weitere Einzelheiten finden Sie in den zugehörigen Informationen.

Wenn es für Sie nicht wichtig ist, ob TLS-Zertifikate widerrufen werden, da Sie das Programm vielleicht in einer Testumgebung ausführen, können Sie 'OCSPCheckExtensions' in der SSL-Zeilengruppe auf NO setzen. Wenn Sie diese Variable festlegen, wird jede AIA-Zertifikatserweiterung ignoriert. Diese Lösung ist in einer Produktionsumgebung wahrscheinlich nicht akzeptabel, da Sie wahrscheinlich nicht den Zugriff von Benutzern mit widerrufbaren Zertifikaten zulassen möchten.

Der Aufruf zum Zugriff auf den OCSP-Responder kann zu einem der folgenden drei Ergebnisse führen:

Good (Gut)
Das Zertifikat ist gültig.

Revoked (Widerrufen)
Das Zertifikat wird entzogen.

Unbekannt
Dieses Ergebnis kann sich aus einem der drei folgenden Gründe ergeben:

- IBM MQ kann nicht auf den OCSP-Responder zugreifen.
- Der OCSP-Responder hat eine Antwort gesendet, IBM MQ kann die digitale Signatur der Antwort jedoch nicht überprüfen.
- Der OCSP-Responder hat eine Antwort gesendet, die anzeigt, dass sie keine Widerrufsdaten für das Zertifikat hat.

Wenn IBM MQ das OCSP-Ergebnis `Unbekannt` empfängt, hängt sein Verhalten von der Einstellung des Attributs 'OCSPAuthentication' ab. Bei WS-Managern wird dieses Attribut an einer der folgenden Positionen gehalten:

- Linux AIX In der SSL-Zeilengruppe der `qm.ini`-Datei unter AIX and Linux.
- Windows In der Windows-Registry.

Dieses Attribut kann mit dem IBM MQ Explorer festgelegt werden. Für Clients wird das Attribut in der SSL-Zeilengruppe der Clientkonfigurationsdatei gehalten.

Wenn das Ergebnis `Unbekannt` empfangen wird und 'OCSPAuthentication' auf REQUIRED gesetzt ist (Standardwert), lehnt IBM MQ die Verbindung ab und gibt eine Fehlernachricht vom Typ AMQ9716 aus. Wenn WS-Manager-SSL-Ereignisnachrichten aktiviert sind, wird eine SSL-Ereignisnachricht vom Typ MQRC_CHANNEL_SSL_ERROR mit dem Wert MQRQ_SSL_HANDSHAKE_ERROR generiert, die auf MQRQ_SSL_HANDSHAKE_ERROR gesetzt ist.

Wenn das Ergebnis `Unbekannt` empfangen wird und 'OCSPAuthentication' auf OPTIONAL gesetzt ist, ermöglicht IBM MQ den Start des SSL-Kanals und es werden keine Warnungen oder SSL-Ereignisnachrichten generiert.

Wenn das Ergebnis `Unbekannt` empfangen wird und 'OCSPAuthentication' auf WARN gesetzt ist, startet der SSL-Kanal, IBM MQ gibt aber einen Warnhinweis vom Typ AMQ9717 im Fehlerprotokoll aus. Wenn WS-Manager-SSL-Ereignisnachrichten aktiviert sind, wird eine SSL-Ereignisnachricht vom

Typ MQRC_CHANNEL_SSL_WARNING mit dem auf MQRQ_SSL_UNKNOWN_REVOCATION gesetzten ReasonQualifier-Set generiert.

Digitale Signatur von OCSP-Antworten

Ein OCSP-Responder kann seine Antworten auf eine von drei Arten signieren. Ihr Responder informiert Sie darüber, welche Methode verwendet wird.

- Die OCSP-Antwort kann mit einem CA-Zertifikat signiert werden, das das Zertifikat ausgestellt hat, das Sie überprüfen. In diesem Fall müssen Sie kein zusätzliches Zertifikat einrichten. Die Schritte, die Sie bereits zur Einrichtung der TLS-Konnektivität unternommen haben, reichen aus, um die OCSP-Antwort zu überprüfen.
- Die OCSP-Antwort kann digital signiert werden, indem ein anderes Zertifikat signiert wird, das von derselben Zertifizierungsstelle (CA) signiert wurde, die das Zertifikat ausgestellt hat, das Sie überprüfen. Das Signaturzertifikat wird in diesem Fall zusammen mit der OCSP-Antwort gesendet. Für das Zertifikat, das vom OCSP-Responder aus ausgeführt wurde, muss die Erweiterung "Extended Key Usage" auf `id-kp-OCSPSigning` gesetzt sein, damit es für diesen Zweck vertrauenswürdig ist. Da die OCSP-Antwort mit dem signierten Zertifikat gesendet wird (und das Zertifikat von einer CA signiert wird, die bereits für TLS-Konnektivität anerkannt ist), ist keine zusätzliche Zertifikatskonfiguration erforderlich.
- Die OCSP-Antwort kann digital signiert werden, indem ein anderes Zertifikat verwendet wird, das nicht direkt mit dem Zertifikat verknüpft ist, das Sie überprüfen. In diesem Fall wird die OCSP-Antwort durch ein Zertifikat signiert, das vom OCSP-Responder selbst ausgestellt wurde. Der Schlüsseldatenbank des Clients oder Warteschlangenmanagers, der die OCSP-Prüfung vornimmt, muss eine Kopie des OCSP-Responderzertifikats hinzugefügt werden. Informationen hierzu finden Sie unter „CA-Zertifikat oder öffentlichen Teil eines selbst signierten Zertifikats unter AIX, Linux, and Windows einem Schlüsselrepository hinzufügen" auf Seite 327. Wenn ein CA-Zertifikat hinzugefügt wird, wird es standardmäßig als Trusted Root hinzugefügt. Dies ist die erforderliche Einstellung in diesem Kontext. Wird dieses Zertifikat nicht hinzugefügt, kann IBM MQ die digitale Signatur in der OCSP-Antwort nicht überprüfen und die OCSP-Prüfung führt zu einem unbekannten Ergebnis, wodurch IBM MQ den Kanal je nach dem Wert von OCSPAuthentication möglicherweise schließt.

Online Certificate Status Protocol (OCSP) in Java und JMS-Clientanwendungen

Aufgrund einer Einschränkung der Java API kann IBM MQ die OCSP-Überprüfung (Online Certificate Status Protocol) für die Zertifikatswiderrufsprüfung für TLS Secure Sockets nur verwenden, wenn OCSP für den gesamten JVM-Prozess (Java Virtual Machine) aktiviert ist. Es gibt zwei Möglichkeiten, OCSP für alle sicheren Sockets in der JVM zu aktivieren:

- Bearbeiten Sie die JRE-Datei 'java.security', um die OCSP-Konfigurationseinstellungen einzuschließen, die in Tabelle 1 aufgeführt sind, und starten Sie die Anwendung erneut.
- Verwenden Sie die API 'java.security.Security.setProperty()', falls eine Java Security Manager-Richtlinie gültig ist.

Als Mindestwert müssen Sie einen der Werte ocsp.enable und ocsp.responderURL angeben.

Eigenschaftsname	Beschreibung
ocsp.enable	Der Wert dieser Eigenschaft ist entweder `true` oder `false` . Wenn `true` aktiviert ist, wird die OCSP-Prüfung aktiviert, wenn die Zertifikatswiderrufsprüfung durchgeführt wird. Wenn `false` oder nicht festgelegt ist, ist die OCSP-Prüfung inaktiviert.
ocsp.responderURL	Der Wert dieser Eigenschaft ist eine URL, die die Position des OCSP-Responder angibt. Hier ein Beispiel: `ocsp.responderURL=http://ocsp.example.net:80`. Standardmäßig wird die Position des OCSP-Responders implizit aus dem Zertifikat ermittelt, das geprüft wird. Die Eigenschaft wird verwendet, wenn die Erweiterung "Berechtigung Information Access" (die in RFC 3280 definiert ist) nicht im Zertifikat vorhanden ist oder wenn sie überschrieben werden muss.

Eigenschaftsname	Beschreibung
ocsp.responderCertSubjectName	Der Wert dieses Merkmals ist der Betreffname des Zertifikats des OCSP-Responders. Hier ein Beispiel: `ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"`. Standardmäßig ist das Zertifikat des OCSP-Responders der des Ausstellers des Zertifikats, das geprüft wird. Diese Eigenschaft gibt das Zertifikat des OCSP-Responders an, wenn der Standardwert nicht angewendet wird. Sein Wert ist ein definierter Zeichenfolgename (in RFC 2253 definiert), der ein Zertifikat in der Gruppe von Zertifikaten angibt, die während der Validierung des Zertifikatpfads bereitgestellt werden. In den Fällen, in denen der Betreffname allein nicht ausreicht, um das Zertifikat eindeutig zu identifizieren, müssen stattdessen die Merkmale ocsp.responderCertIssuerName und ocsp.responderCertSerialNumber verwendet werden. Wenn diese Eigenschaft gesetzt ist, werden die Eigenschaften ocsp.responderCertIssuerName und ocsp.responderCertSerialNumber ignoriert.
ocsp.responderCertIssuerName	Der Wert dieser Eigenschaft ist der Name des Ausstellers des OCSP-Responder-Zertifikats. Hier ein Beispiel: `ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"`. Standardmäßig ist das Zertifikat des OCSP-Responders der des Ausstellers des Zertifikats, das geprüft wird. Diese Eigenschaft gibt das Zertifikat des OCSP-Responders an, wenn der Standardwert nicht angewendet wird. Sein Wert ist ein definierter Zeichenfolgename (in RFC 2253 definiert), der ein Zertifikat in der Gruppe von Zertifikaten angibt, die während der Validierung des Zertifikatpfads bereitgestellt werden. Wenn diese Eigenschaft festgelegt wird, muss auch die Eigenschaft 'ocsp.responderCertSerialNumber' festgelegt werden. Diese Eigenschaft wird ignoriert, wenn die Eigenschaft 'ocsp.responderCertSubjectName' festgelegt ist.
ocsp.responderCertSerialNumber	Bei diesem Wert handelt es sich um die Seriennummer des Zertifikats des OCSP-Responderers. Hier ein Beispiel: `ocsp.responderCertSerialNumber=2A:FF:00`. Standardmäßig ist das Zertifikat des OCSP-Responders der des Ausstellers des Zertifikats, das geprüft wird. Diese Eigenschaft gibt das Zertifikat des OCSP-Responders an, wenn der Standardwert nicht angewendet wird. Dieser Wert ist eine Zeichenfolge aus Hexadezimalziffern (Doppelpunkt-oder Leerzeichen-Trennzeichen), die ein Zertifikat in der Gruppe von Zertifikaten identifizieren, die während der Validierung des Zertifikatpfads bereitgestellt werden. Wenn diese Eigenschaft festgelegt wird, muss auch die Eigenschaft 'ocsp.responderCertIssuerName' festgelegt werden. Diese Eigenschaft wird ignoriert, wenn die Eigenschaft 'ocsp.responderCertSubjectName' festgelegt ist.

Bevor Sie OCSP auf diese Weise aktivieren, gibt es eine Reihe von Überlegungen:

- Das Festlegen der OCSP-Konfiguration wirkt sich auf alle sicheren Sockets im JVM-Prozess aus. In einigen Fällen kann diese Konfiguration unerwünschte Nebeneffekte haben, wenn die JVM mit einem anderen Anwendungscode, der TLS-Secure Sockets verwendet, gemeinsam genutzt wird. Stellen Sie sicher, dass die ausgewählte OCSP-Konfiguration für alle Anwendungen geeignet ist, die in derselben JVM ausgeführt werden.
- Wenn Sie die Wartung auf Ihre JRE anwenden, wird möglicherweise die Datei "java.security" überschrieben. Achten Sie bei der Anwendung von vorläufigen Fixes und der Produktwartung für Java darauf, dass die Datei 'java.security' nicht überschrieben wird. Es kann erforderlich sein, Ihre java.security-Änderungen erneut anzuwenden, nachdem Sie die Wartung angewendet haben. Aus diesem Grund können Sie die OCSP-Konfiguration möglicherweise mit der API "java.security.Security.setProperty ()" definieren.
- Die Aktivierung der OCSP-Prüfung wirkt sich nur dann aus, wenn die Widerrufsprüfung ebenfalls aktiviert ist. Die Widerrufsprüfung wird durch die `PKIXParameters.setRevocationEnabled()` -Methode aktiviert.
- Wenn Sie den AMS Java-Interceptor verwenden, der unter OCSP-Prüfung in nativen Interceptors aktivieren beschrieben ist, müssen Sie sicherstellen, dass Sie in der OCSP-Konfiguration keine java.security verwenden, die mit der AMS-OCSP-Konfiguration in der Konfigurationsdatei des Keystores in Konflikt steht.

Mit Zertifikatswiedergabelisten und Berechtigungslisten für die Berechtigung arbeiten

Die IBM MQ-Unterstützung für CRLs und ARLs ist von der Plattform abhängig.

Die CRL-und ARL-Unterstützung auf jeder Plattform ist wie folgt:

- Unter z/OS unterstützt System SSL die CRLs und ARLs, die vom Tivoli Public Key Infrastructure-Produkt in den LDAP-Servern gespeichert wurden.
- Auf anderen Plattformen entspricht die CRL- und ARL-Unterstützung den Empfehlungen für PKIX-CRL-Profile gemäß Standard X.509 V2.

IBM MQ verwaltet einen Cache mit CRLs und ARLs, auf die in den letzten 12 Stunden zugegriffen wurde.

Wenn ein Warteschlangenmanager oder ein IBM MQ MQI client ein Zertifikat empfängt, wird anhand der CRL geprüft, ob das Zertifikat noch gültig ist. IBM MQ überprüft zunächst den Cache, falls einer vorhanden ist. Wenn sich die CRL nicht im Cache befindet, fragt IBM MQ die LDAP-CRL-Server-Positionen in der Reihenfolge ab, in der sie in der Namensliste der Authentifizierungsinformationsobjekte erscheinen, die durch das Attribut *SSLCRLNL* angegeben wurden, bis IBM MQ eine verfügbare CRL findet. Wenn die Namensliste nicht angegeben ist oder mit einem Leerwert angegeben wird, werden CRLs nicht überprüft.

LDAP-Server einrichten

Konfigurieren Sie die Struktur des LDAP-Verzeichnisinformationsbaums so, dass sie die Hierarchie der definierten Namen von CAs wiedergibt. Verwenden Sie dazu die Dateien des LDAP-Dateninterchange-Formats.

Konfigurieren Sie die Struktur des LDAP-Verzeichnisinformationsbaums (LDAP Directory Information Tree, DIT) so, dass die Hierarchie verwendet wird, die den definierten Namen der CAs entspricht, die die Zertifikate und Zertifikatswiderrufslisten ausgeben. Sie können die DIT-Struktur mit einer Datei konfigurieren, die das LDAP-Dateninterchange-Format (LDIF) verwendet. Sie können auch LDIF-Dateien verwenden, um ein Verzeichnis zu aktualisieren.

Bei LDIF-Dateien handelt es sich um ASCII-Textdateien, die die Informationen enthalten, die zum Definieren von Objekten in einem LDAP-Verzeichnis erforderlich sind. LDIF-Dateien enthalten einen oder mehrere Einträge, die jeweils einen definierten Namen (Distinguished Name), mindestens eine Objektklassendefinition und optional mehrere Attributdefinitionen enthalten.

Das Attribut `certificateRevocationList;binary` enthält eine Liste der widerruften Benutzerzertifikate in binärer Form. Das Attribut `authorityRevocationList;binary` enthält eine binäre Liste von CA-Zertifikaten, die widerrufen wurden. Für die Verwendung mit IBM MQ-TLS müssen die Binärdaten für diese Attribute dem DER-Format (Definite Encoding Rules) entsprechen. Weitere Informationen zu LDIF-Dateien finden Sie in der Dokumentation, die mit dem LDAP-Server bereitgestellt wird.

Abbildung 20 auf Seite 369 zeigt eine LDIF-Beispieldatei, die Sie als Eingabe für Ihren LDAP-Server erstellen können, um die von CA1 ausgegebenen CRLs und ARLs zu laden. Es handelt sich hierbei um eine fiktive Zertifizierungsstelle mit dem definierten Namen "CN=CA1, OU=Test, O=IBM, C=GB", die von der Prüforganisation von IBM eingerichtet wurde.

```
dn: o=IBM, c=GB
o: IBM
objectclass: top
objectclass: organization

dn: ou=Test, o=IBM, c=GB
ou: Test
objectclass: organizationalUnit

dn: cn=CA1, ou=Test, o=IBM, c=GB
cn: CA1
objectclass: cRLDistributionPoint
objectclass: certificateAuthority
authorityRevocationList;binary:: (DER format data)
certificateRevocationList;binary:: (DER format data)
caCertificate;binary:: (DER format data)
```

Abbildung 20. LDIF-Beispieldatei für eine Zertifizierungsstelle. Dies kann von der Implementierung bis zur Implementierung variieren.

Abbildung 21 auf Seite 369 zeigt die DIT-Struktur, die Ihr LDAP-Server erstellt, wenn Sie die in Abbildung 20 auf Seite 369 gezeigte LDIF-Beispieldatei zusammen mit einer ähnlichen CA2-Datei (eine weitere fiktive Zertifizierungsstelle, die von der PKI-Organisation in IBM eingerichtet wurde) laden.

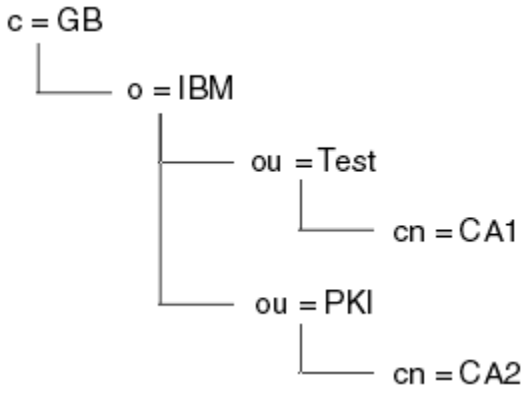


Abbildung 21. Beispiel für eine Struktur des LDAP-Verzeichnisinformationsbaums

IBM MQ überprüft sowohl CRLs als auch ARLs.

Anmerkung: Stellen Sie sicher, dass die Zugriffssteuerungsliste für Ihren LDAP-Server berechtigt ist, die Einträge zu lesen, zu suchen und zu vergleichen, die die CRLs und ARLs enthalten. IBM MQ greift über die Eigenschaften LDAPUSER und LDAPPWD des AUTHINFO-Objekts auf den LDAP-Server zu.

LDAP-Server konfigurieren und aktualisieren
Gehen Sie zur Konfiguration und Aktualisierung des LDAP-Servers wie hier beschrieben vor.

1. Fordern Sie von Ihrer Zertifizierungsstelle bzw. Ihren Zertifizierungsstellen die Zertifikatssperrlisten und CA-Zertifikatssperrlisten im DER-Format an.
2. Erstellen Sie mit einem Texteditor oder einem im LDAP-Server verfügbaren Tool eine oder mehrere LDIF-Dateien, die den definierten Namen der Zertifizierungsstelle sowie die erforderlichen Objektklassendefinitionen enthalten. Kopieren Sie die Daten im DER-Format als Werte für das Attribut `certificateRevocationList;binary` (CRLs) und/oder für das Attribut `authorityRevocationList;binary` (ARLs) in die LDIF-Datei.
3. Starten Sie den LDAP-Server.
4. Fügen Sie die Einträge aus der unter Schritt „2” auf Seite 369 erstellten LDIF-Datei hinzu.

Überprüfen Sie den LDAP-CRL-Server im Anschluss an die Konfiguration. Verwenden Sie zunächst ein Zertifikat, das auf dem Kanal nicht gesperrt ist, und vergewissern Sie sich, dass der Kanal korrekt gestartet

wird. Verwenden Sie anschließend ein gesperrtes Zertifikat, und vergewissern Sie sich, dass der Kanal nicht gestartet wird.

Sie sollten Zertifikatssperrlisten so oft wie möglich von Zertifizierungsstellen anfordern. Auf Ihren LDAP-Servern sollte dies alle 12 Stunden erfolgen.

Zugriff auf CRLs und ARLs mit einem WS-Manager

Ein WS-Manager ist einem oder mehreren Authentifizierungsinformationsobjekten zugeordnet, die die Adresse eines LDAP-CRL-Servers enthalten. IBM i IBM MQ unter IBM i verhält sich anders als auf anderen Plattformen.

Beachten Sie, dass in diesem Abschnitt auch Informationen zu Zertifikatswiderrufslisten (Certificate Revocation Lists, CRLs) für die Berechtigungswiderrufslisten (ARLs, Authority Revocation Lists) gelten.

Sie sagen dem Warteschlangenmanager, wie auf CRLs zugegriffen werden kann, indem er den Warteschlangenmanager mit Authentifizierungsinformationsobjekten versorgt, die jeweils die Adresse eines LDAP-CRL-Servers enthalten. Die Authentifizierungsinformationsobjekte werden in einer Namensliste gespeichert, die im Warteschlangenmanager-Attribut *SSLCRLNL* angegeben ist.

Im folgenden Beispiel wird MQSC verwendet, um die Parameter anzugeben:

1. Definieren Sie Authentifizierungsinformationsobjekte mit dem MQSC-Befehl DEFINE AUTHINFO, wobei der Parameter AUTHTYPE auf CRLLDAP gesetzt ist. IBM i Unter IBM i können Sie auch den Befehl CRTMQMAUTI CL verwenden.

   Der Wert CRLLDAP für den Parameter AUTHTYPE gibt an, dass auf LDAP-Server auf CRLs zugegriffen wird. Jedes Authentifizierungsinformationsobjekt mit dem Typ CRLLDAP, das Sie erstellen, enthält die Adresse eines LDAP-Servers. Wenn Sie mehr als ein Authentifizierungsinformationsobjekt haben, müssen die LDAP-Server, auf die sie verweisen, identische Informationen enthalten. Dies bietet die Kontinuität des Service, wenn ein oder mehrere LDAP-Server fehlschlagen.

   z/OS Nur unter z/OS muss der Zugriff auf alle LDAP-Server mit der gleichen Benutzer-ID und dem gleichen Kennwort erfolgen. Die verwendete Benutzer-ID und das Kennwort sind die Benutzer, die im ersten AUTHINFO-Objekt in der Namensliste angegeben sind.

   Auf allen Plattformen werden die Benutzer-ID und das Kennwort unverschlüsselt an den LDAP-Server gesendet.
2. Definieren Sie mit dem MQSC-Befehl DEFINE NAMELIST eine Namensliste für die Namen Ihrer Authentifizierungsinformationsobjekte. z/OS Stellen Sie unter z/OS sicher, dass das Namenslistenattribut NLTYPE auf AUTHINFO gesetzt ist.
3. Verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie die Namensliste an den Warteschlangenmanager an. Beispiel:

   ```
   ALTER QMGR SSLCRLNL(sslcrlnlname)
   ```

   Hierbei steht `sslcrlnlname` für Ihre Namensliste mit Authentifizierungsinformationsobjekten.

   Mit diesem Befehl wird ein Warteschlangenmanager-Attribut mit dem Namen *SSLCRLNL* festgelegt. Der Anfangswert des WS-Managers für dieses Attribut ist leer.

IBM i Unter IBM i können Sie Authentifizierungsinformationsobjekte angeben, aber der Warteschlangenmanager verwendet weder Authentifizierungsinformationsobjekte noch eine Namensliste mit Authentifizierungsinformationsobjekten. Nur IBM MQ-Clients, die eine Clientverbindungstabelle verwenden, die von einem IBM i-MQ-Queue Manager generiert wird, verwenden die Authentifizierungsinformationen, die für diesen IBM i-Queue Manager angegeben sind. Das Warteschlangenmanagerattribut *SSLCRLNL* in IBM i legt fest, welche Authentifizierungsinformationen von diesen Clients verwendet werden. Im Abschnitt „Zugriff auf CRLs und ARLs unter IBM i" auf Seite 371 finden Sie Informationen dazu, wie Sie einem IBM i -Warteschlangenmanager mitteilen, wie er auf CRLs zugreifen kann.

Sie können bis zu 10 Verbindungen zu alternativen LDAP-Servern zu der Namensliste hinzufügen, um die Kontinuität des Service zu gewährleisten, wenn ein oder mehrere LDAP-Server ausfallen. Beachten Sie, dass die LDAP-Server identische Informationen enthalten müssen.

IBM i *Zugriff auf CRLs und ARLs unter IBM i*

Gehen Sie folgendermaßen vor, um auf CRLs oder ARLs unter IBM i zuzugreifen.

Beachten Sie, dass in diesem Abschnitt auch Informationen zu Zertifikatswiderrufslisten (Certificate Revocation Lists, CRLs) für die Berechtigungswiderrufslisten (ARLs, Authority Revocation Lists) gelten.

Führen Sie die folgenden Schritte aus, um eine CRL-Position für ein bestimmtes Zertifikat unter IBM i zu konfigurieren:

1. Rufen Sie, wie unter „Zugriff auf DCM” auf Seite 294 beschrieben, die DCM-Schnittstelle auf.
2. Klicken Sie in der Taskkategorie **CRL-Standorte verwalten** in der Navigationsanzeige auf **CRL-Position hinzufügen** . Die Seite 'CRL-Standorte verwalten' wird im Taskrahmen angezeigt.
3. Geben Sie im Feld **Name des CRL-Verteilungspunkts** einen Namen für den CRL-Verteilungspunkt ein, z. B. `LDAP Server #1` .
4. Geben Sie in das Feld **LDAP-Server** den Namen des LDAP-Servers ein.
5. Wählen Sie im Feld **Secure Sockets Layer (SSL) verwenden** die Option **Ja** aus, wenn Sie mit TLS eine Verbindung zum LDAP-Server herstellen möchten. Wählen Sie andernfalls **Nein** aus.
6. Geben Sie in das Feld **Portnummer** eine Portnummer für den LDAP-Server ein, z. B. 389.
7. Wenn Ihr LDAP-Server es anonymen Benutzern nicht erlaubt, das Verzeichnis abzufragen, geben Sie im Feld **Anmelde-DN** einen registrierten Namen für den Server ein.
8. Klicken Sie auf **OK**. DCM informiert Sie darüber, dass die CRL-Position erstellt wurde.
9. Klicken Sie in der Navigationsanzeige auf **Select a Certificate Store** (Zertifikatsspeicher auswählen). Die Seite 'Select a Certificate Store' (Zertifikatsspeicher auswählen) wird im Taskrahmen angezeigt.
10. Wählen Sie das Kontrollkästchen **Other System Certificate Store** aus und klicken Sie auf **Continue** . Die Seite Zertifikatsspeicher und Kennwort wird angezeigt.
11. Geben Sie im Feld **Certificate store path and filename** (Pfad und Dateiname des Zertifikatsspeichers) den IFS-Pfad und Dateinamen ein, die Sie im Abschnitt „Zertifikatsspeicher unter IBM i erstellen” auf Seite 296 definiert haben.
12. Geben Sie ein Kennwort in das Feld **Certificate Store Password** ein. Klicken Sie auf **Weiter** . Die Seite Aktuelle Zertifikatsspeicher wird im Taskrahmen angezeigt.
13. Klicken Sie in der Taskkategorie **Manage Certificates** in der Navigationsanzeige auf **CRL-Positionszuordnung aktualisieren** . Die Seite 'CRL-Standortzuordnung' wird im Taskrahmen angezeigt.
14. Wählen Sie das Optionsfeld für das CA-Zertifikat aus, dem Sie die CRL-Position zuordnen möchten. Klicken Sie auf **Update CRL Location Assignment** . Die Seite "CRL-Positionszuordnung aktualisieren" wird im Taskrahmen angezeigt.
15. Wählen Sie den Radioknopf für die CRL-Position aus, die Sie dem Zertifikat zuordnen möchten. Klicken Sie auf **Update Assignment** . DCM informiert Sie darüber, dass die Zuordnung aktualisiert wurde.

Beachten Sie, dass DCM Ihnen die Möglichkeit gibt, einen anderen LDAP-Server von der Zertifizierungsstelle zuzuordnen.

Zugriff auf CRLs und ARLs mithilfe von IBM MQ Explorer

Sie können einem Warteschlangenmanager mithilfe von IBM MQ Explorer mitteilen, wie der Zugriff auf CRLs erfolgt.

Beachten Sie, dass in diesem Abschnitt auch Informationen zu Zertifikatswiderrufslisten (Certificate Revocation Lists, CRLs) für die Berechtigungswiderrufslisten (ARLs, Authority Revocation Lists) gelten.

Gehen Sie wie folgt vor, um eine LDAP-Verbindung zu einer CRL einzurichten:

1. Stellen Sie sicher, dass der WS-Manager gestartet wurde.

2. Klicken Sie mit der rechten Maustaste auf den Ordner **Authentifizierungsinformationen** und klicken Sie auf **Neu -> Authentifizierungsinformationen**. In dem Eigenschaftenblatt, das geöffnet wird:
   a. Geben Sie auf der ersten Seite **Authentifizierungsinformationen erstellen** einen Namen für das CRL-Objekt (LDAP) ein.
   b. Wählen Sie auf der Seite **Allgemein** von **Eigenschaften ändern** den Verbindungstyp aus. Optional können Sie eine Beschreibung eingeben.
   c. Wählen Sie die Seite **CRL (LDAP)** von **Change Properties** (Eigenschaften ändern) aus.
   d. Geben Sie den Namen des LDAP-Servers entweder als Netznamen oder als IP-Adresse ein.
   e. Wenn der Server Anmeldedaten erfordert, stellen Sie eine Benutzer-ID und falls erforderlich ein Kennwort bereit.
   f. Klicken Sie auf **OK**.
3. Klicken Sie mit der rechten Maustaste auf den Ordner Namenslisten und klicken Sie auf **Neu -> Namenslisten**. In dem Eigenschaftenblatt, das geöffnet wird:
   a. Geben Sie einen Namen für die Namensliste ein.
   b. Fügen Sie den unter Schritt „2.a" auf Seite 372 angegebenen Namen für das CRL(LDAP)-Objekt der Liste hinzu.
   c. Klicken Sie auf **OK**.
4. Klicken Sie auf den Warteschlangenmanager, wählen Sie **Eigenschaften** aus, und wählen Sie die Seite **SSL** aus:
   a. Wählen Sie das Kontrollkästchen **Zertifikate überprüfen, die von diesem WS-Manager empfangen werden, in der Liste der Zertifizierungsaufrufslisten** aus.
   b. Geben Sie im Feld **CRL-Namensliste** den unter Schritt „3.a" auf Seite 372 angegebenen Namen der Namensliste ein.

Mit einem IBM MQ MQI client auf CRLs und ARLs zugreifen

Sie haben drei Optionen für die Angabe der LDAP-Server, die CRLs für die Überprüfung durch einen IBM MQ MQI client enthalten.

Beachten Sie, dass in diesem Abschnitt auch Informationen zu Zertifikatswiderrufslisten (Certificate Revocation Lists, CRLs) für die Berechtigungswiderrufslisten (ARLs, Authority Revocation Lists) gelten.

Es gibt die folgenden drei Möglichkeiten, die LDAP-Server anzugeben:

- Verwenden einer Kanaldefinitionstabelle
- Verwendung der SSL-Konfigurationsoptionsstruktur, MQSCO, in einem MQCONNX-Aufruf
- Verwendung von Active Directory (auf Windows-Systemen mit Active Directory-Unterstützung)

Weitere Informationen finden Sie in den zugehörigen Informationen.

Sie können bis zu 10 Verbindungen zu alternativen LDAP-Servern aufnehmen, um die Kontinuität des Service zu gewährleisten, wenn ein oder mehrere LDAP-Server fehlschlagen. Beachten Sie, dass die LDAP-Server identische Informationen enthalten müssen.

Sie können von einem IBM MQ MQI client-Kanal, der auf Linux (zSeries-Plattform) ausgeführt wird, nicht auf LDAP-CRLs zugreifen.

Position eines OCSP-Responders und von LDAP-Servern, die CRLs enthalten
In einem IBM MQ MQI client-System können Sie die Position eines OCSP-Responders und des LDAP-Servers mit den Zertifikatsperrliste (Certificate Revocation Lists, CRLs) angeben.

Sie können diese Positionen auf drei Arten angeben, die hier beschrieben werden, um die Vorrangstellung zu verringern.

IBM i Informationen zu IBM i finden Sie unter Zugriff auf CRLs und ARLs in IBM i.

Wenn eine IBM MQ MQI client-Anwendung einen MQCONNX-Aufruf ausgibt

Sie können einen OCSP-Responder oder einen LDAP-Server mit CRLs in einem **MQCONNX** -Aufruf angeben.

Bei einem **MQCONNX** -Aufruf kann die Struktur der Verbindungsoptionen (MQCNO) auf eine Struktur der SSL-Konfigurationsoptionen (MQSCO) verweisen. Die MQSCO-Struktur kann wiederum auf eine oder mehrere Authentifizierungsdaten-Satzstrukturen (MQAIR) verweisen. Jede MQAIR-Struktur enthält alle Informationen, die ein IBM MQ MQI client für den Zugriff auf einen OCSP-Responder oder einen LDAP-Server mit CRLs benötigt. Beispiel: Eines der Felder in einer MQAIR-Struktur ist die URL, an die ein Responder kontaktiert werden kann. Weitere Informationen zur MQAIR-Struktur finden Sie im Abschnitt MQAIR-Authentication information record .

Verwenden einer Clientkanaldefinitionstabelle (ccdt) für den Zugriff auf einen OCSP-Responder oder LDAP-Server

Damit ein IBM MQ MQI client auf einen OCSP-Responder oder LDAP-Server mit CRLs zugreifen kann, schließen Sie die Attribute eines oder mehrerer Authentifizierungsdatenobjekte in die Definitionstabelle für einen Clientkanal ein.

Auf einem Server-WS-Manager können Sie ein oder mehrere Authentifizierungsinformationsobjekte definieren. Die Attribute eines Authentifizierungsobjekts enthalten alle Informationen, die für den Zugriff auf einen OCSP-Responder (auf Plattformen, auf denen OCSP unterstützt wird) oder ein LDAP-Server, der CRLs enthält, enthalten sind. Eines der Attribute gibt die OCSP-Responder-URL an, eine andere gibt die Hostadresse oder die IP-Adresse eines Systems an, auf dem ein LDAP-Server ausgeführt wird.

z/OS IBM i Ein Authentifizierungsinformationsobjekt mit AUTHTYPE (OCSP) gilt nicht für die Verwendung auf IBM i- oder z/OS-Queue Managern, aber es kann auf diesen Plattformen angegeben werden, um für die Clientverwendung in die Definitionstabelle für Clientkanäle (CCDT) kopiert zu werden.

Um den Zugriff eines IBM MQ MQI clients auf einen OCSP-Responder oder auf LDAP-Server mit CRLs zu aktivieren, können die Attribute eines oder mehrerer Authentifizierungsdatenobjekte in eine Definitionstabelle für den Clientkanal eingeschlossen werden. Sie können solche Attribute auf eine der folgenden Arten einschließen:

Multi

Auf Serverplattformen AIX, Linux, IBM iund Windows

Sie können eine Namensliste definieren, die die Namen von einem oder mehreren Authentifizierungsinformationsobjekten enthält. Anschließend können Sie das Warteschlangenmanagerattribut **SSLCRLNL** auf den Namen dieser Namensliste setzen.

Wenn Sie CRLs verwenden, kann mehr als ein LDAP-Server konfiguriert werden, um eine höhere Verfügbarkeit bereitzustellen. Es wird beabsichtigt, dass jeder LDAP-Server dieselben CRLs enthält. Falls ein LDAP-Server nicht verfügbar ist, wenn er benötigt wird, kann ein an IBM MQ MQI client auf einen anderen LDAP-Server zugreifen.

Die Attribute der Authentifizierungsinformationsobjekte, die von der Namensliste identifiziert werden, werden hier zusammen als *Zertifikatswiderrufsposition* bezeichnet. Wenn Sie das Warteschlangenmanagerattribut **SSLCRLNL** auf den Namen der Namensliste setzen, wird die Zertifikatswiderrufungsposition in die Client-Kanaldefinitionstabelle kopiert, die dem Warteschlangenmanager zugeordnet ist. Wenn der Zugriff auf die CCDT über ein Clientsystem als gemeinsam genutzte Datei möglich ist, oder wenn die CCDT dann in ein Clientsystem kopiert wird, kann der IBM MQ MQI client auf diesem System die Position für den Zertifikatswiderruf in der CCDT verwenden, um auf einen OCSP-Responder oder auf LDAP-Server mit CRLs zuzugreifen.

Wenn die Zertifikatswiderrufposition des WS-Managers später geändert wird, wird die Änderung in der CCDT wiedergegeben, die dem Warteschlangenmanager zugeordnet ist. Wenn das Warteschlangenmanagerattribut **SSLCRLNL** auf „leer“ gesetzt ist, wird die Zertifikatswiderrufungsposition aus der CCDT entfernt. Diese Änderungen werden in keiner Kopie der Tabelle auf einem Clientsystem widergespiegelte Änderungen.

Wenn die Zertifikatswiderrufsposition auf dem Client-und dem Serverende eines MQI-Kanals unterschiedlich sein muss und der Server-WS-Manager der Name des Servers ist, der zum Erstellen der Zertifikatswiderrufsposition verwendet wird, können Sie die Zertifikatswiderrufsposition wie folgt ausführen:

1. Erstellen Sie auf dem Server-WS-Manager die Zertifikatswiderrufsposition für die Verwendung auf dem Clientsystem.
2. Kopieren Sie die CCDT, die die Position des Zertifikatswiderrufs enthält, auf das Clientsystem.
3. Ändern Sie auf dem Server-WS-Manager die Zertifikatswiderrufsposition in die Angabe, die am Serverende des MQI-Kanals erforderlich ist.
4. Auf der Clientmaschine können Sie den Befehl **runmqsc** mit dem Parameter **-n** verwenden.

Multi

Auf Clientplattformen AIX, Linux, IBM iund Windows

Sie können eine CCDT auf der Clientmaschine erstellen, indem Sie den Befehl runmqsc mit dem Parameter **-n** und **DEFINE AUTHINFO** -Objekten in der CCDT-Datei verwenden. Die Reihenfolge, in der die Objekte definiert sind, ist die Reihenfolge, in der sie in der Datei verwendet werden. Jeder Name, den Sie möglicherweise in einem **DEFINE AUTHINFO** -Objekt verwenden, wird nicht in der Datei beibehalten. Nur positionsgebundene Zahlen werden verwendet, wenn Sie **DISPLAY** die **AUTHINFO** -Objekte in einer CCDT-Datei verwenden.

Anmerkung: Wenn Sie den Parameter **-n** angeben, dürfen Sie keinen anderen Parameter angeben.

Active Directory unter Windows verwenden

Windows

Auf Windows-Systemen können Sie den Steuerbefehl **setmqcrl** verwenden, um die aktuellen CRL-Informationen in Active Directory zu veröffentlichen.

Befehl **setmqcrl** veröffentlicht keine OCSP-Informationen.

Informationen zu diesem Befehl und seiner Syntax finden Sie in setmqcrl .

Mit IBM MQ classes for Java und IBM MQ classes for JMS auf CRLs und ARLs zugreifen

In IBM MQ classes for Java und IBM MQ classes for JMS wird auf CRLs anders zugegriffen als auf anderen Plattformen.

Weitere Informationen zum Arbeiten mit CRLs und ARLs mit IBM MQ classes for Java finden Sie unter Zertifikatswiderrufslisten verwenden

Weitere Informationen zum Arbeiten mit CRLs und ARLs mit IBM MQ classes for JMS finden Sie unter Objekteigenschaft SSLCERTSTORES

Authentifizierungsinformationsobjekte bearbeiten

Sie können Authentifizierungsdatenobjekte mithilfe von MQSC- oder PCF-Befehlen oder dem IBM MQ Explorer bearbeiten.

Die folgenden MQSC-Befehle wirken sich auf Authentifizierungsinformationsobjekte aus:

- AUTHINFO DEFINIER
- ALTER AUTHINFO
- DELETE AUTHINFO
- DISPLAY AUTHINFO

Eine vollständige Beschreibung dieser Befehle finden Sie in MQSC-Befehle .

Die folgenden programmierbaren Befehlsformat-Befehle (PCF = Programmable Command Format) dienen zur Verarbeitung von Authentifizierungsinformationsobjekten:

- Authentifizierungsinformationen erstellen
- Authentifizierungsinformationen kopieren
- Authentifizierungsinformationen ändern
- Authentifizierungsinformationen löschen
- Authentifizierungsinformationen abgefragt
- Namen der Authentifizierungsinformationen abgefragt

Eine vollständige Beschreibung dieser Befehle finden Sie im Abschnitt Definitionen der programmierbaren Befehlsformate .

Es kann auch der IBM MQ Explorer auf den Plattformen verwendet werden, auf denen er verfügbar ist.

Linux AIX Verwenden der Pluggable Authentication Method (PAM)

Sie können PAM nur auf AIX and Linux-Plattformen verwenden. Ein typisches AIX- oder Linux-System verfügt über PAM-Module, die das traditionelle Authentifizierungsverfahren implementieren. Es kann jedoch mehr geben. Neben der grundlegenden Task zur Validierung von Kennwörtern können PAM-Module auch aufgerufen werden, um zusätzliche Regeln auszuführen.

Konfigurationsdateien definieren, welche Authentifizierungsmethode für die einzelnen Anwendungen verwendet werden soll. Beispielanwendungen sind die Standardterminalanmelde-, ftp-und telnet-Datenstationsanwendungen

Der Vorteil von PAM besteht darin, dass die Anwendung nicht wissen muss, wie die Benutzer-ID tatsächlich authentifiziert wird, oder darauf achten, wie die Benutzer-ID tatsächlich authentifiziert wird. Solange die Anwendung eine korrekte Form von Authentifizierungsdaten für PAM bereitstellen kann, ist der Mechanismus hinter diesem PAM transparent.

Die Form der Authentifizierungsdaten hängt von dem verwendeten System ab. IBM MQ ruft beispielsweise ein Kennwort über Parameter ab, wie die MQCSP-Struktur, die im API-Aufruf MQCONNX verwendet wird.

Wichtig: Sie können das Attribut **AUTHENMD** erst festlegen, wenn Sie IBM MQ 8.0.0 Fix Pack 3 installiert und anschließend den Warteschlangenmanager mit **-e CMDLEVEL=***Ebene* von *802* (im Befehl strmqm) starten, um die erforderliche Befehlsebene festzulegen.

System für die Verwendung von PAM konfigurieren

Der von IBM MQ verwendete Servicename beim Aufruf von PAM lautet *ibmmq*.

Beachten Sie, dass eine IBM MQ-Installation versucht, eine PAM-Standardkonfiguration beizubehalten, die Verbindungen von Betriebssystembenutzern zulässt, die auf bekannten Standardwerten für die verschiedenen Betriebssysteme basieren.

Ihr Systemadministrator muss jedoch überprüfen, ob die in den Dateien `/etc/pam.conf` oder `/etc/pam.d/ibmmq` definierten Regeln immer noch angemessen sind.

Autorisieren des Zugriffs auf Objekte

Dieser Abschnitt enthält Informationen zur Verwendung des Objektberechtigungsmanagers und des Kanalexitprogramms, um den Zugriff auf Objekte zu steuern.

ALW Auf Systemen mit AIX, Linux, and Windows. Sie können den Zugriff auf Objekte steuern, indem Sie den Objektberechtigungsmanager (OAM) verwenden. Diese Themensammlung enthält Informationen zur Verwendung der Befehlsschnittstelle für den OAM.

Dieser Abschnitt enthält außerdem eine Prüfliste, mit der Sie ermitteln können, welche Tasks ausgeführt werden müssen, um die Sicherheit auf Ihrem System auf allen Plattformen anzuwenden, und Hinweise, um Benutzern die Berechtigung zum Verwalten von IBM MQ und die Arbeit mit IBM MQ-Objekten zu erteilen.

Wenn die bereitgestellten Sicherheitsmechanismen Ihre Anforderungen nicht erfüllen, können Sie eigene Kanalexitprogramme entwickeln.

Bestimmen, welcher Benutzer für die Berechtigung verwendet wird

Berechtigungen für den Zugriff auf Ressourcen werden Gruppen erteilt, zu denen der Benutzer gehört, oder in bestimmten Modi direkt dem Benutzer, der der Verbindung zugeordnet ist. Während des Verbindungsprozesses und insbesondere für ferne (Client-) Verbindungen könnte diese Identität durch die Konfiguration des Warteschlangenmanagers geändert werden. Auf dieser Seite werden die verschiedenen Features von IBM MQ und ihre Konfigurationsoptionen aufgelistet, die sich auf die Identität einer verbindenden Anwendung auswirken können, sowie die Reihenfolge, in der diese Features wirksam werden.

Funktionen, die ändern können, welcher Benutzer übernommen wird

Die verschiedenen Funktionen, die festlegen können, welcher Benutzer berechtigt werden soll, lauten wie folgt:

Von der Anwendung bestätigter Benutzer
Wenn eine Fernverbindung von IBM MQgestartet wird, wird der Betriebssystembenutzer, unter dem der Prozess ausgeführt wird, an den empfangenden Warteschlangenmanager gesendet. Dieser Benutzer wird gesendet, um sicherzustellen, dass ein Benutzer für die Berechtigungsprüfung verwendet werden kann, wenn keine weitere Konfiguration vorhanden ist, die den Benutzer ändert.

Es wird nicht empfohlen, diesen Benutzer als Basis für die Berechtigung zu verwenden, da Verbindungen ihre Identität ohne serverseitige Validierung zusichern können. Dies kann sogar den Benutzer mit Verwaltungsaufgaben ('mqm ') umfassen.

MCAUSER-Kanaleinstellung
Anwendungen, die Verbindungen über Netzbindungen herstellen, verwenden dazu eine IBM MQ -Kanaldefinition. Kanaldefinitionen unterstützen das Attribut **MCAUSER** , das verwendet werden kann, um einen anderen Benutzer anzugeben, der für die Berechtigung verwendet werden soll, anstatt den Benutzer anzugeben, der von den verbundenen Anwendungen bestätigt wird.

Verbindungsauthentifizierung ADOPTCTX
Anwendungen können einen Benutzer und ein Kennwort angeben, die zu Authentifizierungszwecken an einen Warteschlangenmanager gesendet werden. Diese Berechtigungsnachweise werden mit der Konfiguration authentifiziert, die für die Verbindungsauthentifizierungsfunktion angegeben ist. Die Option **ADOPTCTX** für die Verbindungsauthentifizierung steuert, ob ein Benutzer für die Berechtigung verwendet werden soll, nachdem er erfolgreich validiert wurde. Wenn der Wert auf YESgesetzt ist, wird der für die Authentifizierung angegebene Benutzer für Berechtigungsprüfungen übernommen.

Kanalauthentifizierungsdatensatz MCAUSER
Während der Verbindungsverarbeitung versucht der Warteschlangenmanager, einen Kanalauthentifizierungsdatensatz zu finden, der der Verbindung entspricht. Wenn ein Kanalauthentifizierungsdatensatz übereinstimmt und sein Attributwert **USERSRC** auf MAPgesetzt ist, ändert IBM MQ den für Berechtigungen verwendeten Benutzer in den Wert des Attributs **MCAUSER** .

Sicherheitsexits
Sicherheitsexits sind angepasste Funktionen, die während der IBM MQ -Sicherheitsverarbeitung geschrieben und aufgerufen werden. Wenn die Funktion aufgerufen wird, wird sie mit einer Kopie der MQCD-Struktur bereitgestellt, die mehrere Felder enthält, die sich auf den Verbindungsbenutzer beziehen, der für Berechtigungsprüfungen verwendet wird. Sicherheitsexits können diese Felder ändern, um den Benutzer zu ändern, der berechtigt wird.

Vorrangregelung

Die folgende Tabelle zeigt die Rangfolge für jede Sicherheitsfunktion, die in „Funktionen, die ändern können, welcher Benutzer übernommen wird” auf Seite 376 beschrieben wird, wenn IBM MQ einen zu berechtigenden Benutzer auswählt. Die Reihenfolge ist von der niedrigsten zur höchsten, d. h., eine Sicherheitsfunktion, die einen Benutzer in der ersten Zeile festlegt, wird durch eine der anderen Zeilen überschrieben.

Tabelle 68. Vorrangregelung für Sicherheitsfunktionen	
Reihenfolge	**Funktion**
1 (niedrigste)	Anwendungszusicherungs-ID
2	Kanaldefinition **MCAUSER** , Attribut
3	Verbindungsauthentifizierung mit **ADOPTCTX(YES)**
4	Kanalauthentifizierungsdatensätze mit **USERSRC(MAP)**
5 (höchste)	Sicherheitsexit

Auswirkungen einer frühzeitigen Übernahme

Verbindungsauthentifizierungs-und Kanalauthentifizierungsdatensätze bieten eine Konfigurationsoption, die steuert, wann die Benutzerakzeptanz für die Verbindungsauthentifizierung ausgeführt wird. Diese Einstellung wird als frühe Übernahme bezeichnet. Wenn die frühzeitige Übernahme aktiviert ist, erfolgt die Übernahme der Verbindungsauthentifizierungsidentität vor der Verarbeitung der Kanalauthentifizierungsdatensätze (d. h., die Kanalauthentifizierungsdatensätze überschreiben alle **CONNAUTH** -Überarbeitungen).

Wenn diese Option inaktiviert ist, wird die Reihenfolge umgekehrt, d. h., Kanalauthentifizierungsdatensätze werden verarbeitet, bevor **CONNAUTH** übernommen wird. In dieser Situation hat die Übernahme der Verbindungsauthentifizierung eine höhere effektive Priorität als Kanalauthentifizierungsdatensätze.

Die Standardeinstellung für die frühe Übernahme ist `enabled`.

ALW Zugriff auf Objekte mithilfe des OAM unter AIX, Linux, and Windows steuern

Der Objektberechtigungsmanager (Object Authority Manager, OAM) stellt eine Befehlsschnittstelle zur Verfügung, mit der die Berechtigung für IBM MQ-Objekte erteilt und widerrufen werden kann.

Zur Verwendung dieser Befehle benötigen Sie die entsprechenden Berechtigungen (siehe „Berechtigung für die Verwaltung von IBM MQ unter AIX, Linux, and Windows” auf Seite 428). Benutzer-IDs, die für die Verwaltung von IBM MQ berechtigt sind, verfügen über die Berechtigung *super user* für den Warteschlangenmanager, d. h., Sie müssen diesen Benutzern keine weiteren Berechtigungen für die Ausgabe von MQI-Aufrufen oder -Befehlen erteilen.

Linux AIX Benutzerbasierte OAM-Berechtigungen unter AIX and Linux

Ab IBM MQ 8.0 kann der Objektberechtigungsmanager (Object Authority Manager; OAM) auf Systemen mit UNIX and Linux sowohl benutzerbasierte als auch gruppenbasierte Berechtigungen verwenden.

Vor IBM MQ 8.0 basieren die Zugriffssteuerungslisten (Access Control Lists, ACLs) unter UNIX and Linux nur auf Gruppen. In IBM MQ 8.0 basieren ACLs auf Benutzer-IDs und Gruppen, und Sie können entweder das benutzerbasierte Modell oder das gruppenbasierte Modell für die Autorisierung verwenden, indem Sie das Attribut **SecurityPolicy** auf den entsprechenden Wert setzen, wie im Abschnitt Installierbare Services konfigurieren und Zeilengruppen für Berechtigungsservices unter AIX and Linux beschrieben.

Änderungen im Verhalten für IBM MQ 8.0 und höher

Ab IBM MQ 8.0 geben einige Befehle bei der Ausführung mit der benutzerbasierten Richtlinie möglicherweise andere Informationen als bei der Verwendung von früheren Versionen des Produkts zurück:

- Die Befehle **dmpmqaut** und **dmpmqcfg** zeigen benutzerabhängige Datensätze an, ebenso wie die PCF-äquivalenten Operationen.

- Das OAM-Plug-in für IBM MQ Explorer zeigt benutzerbasierte Datensätze an und ermöglicht benutzerbasierte Änderungen.
- Die OAM-Funktion **Inquire** gibt Ergebnisse zurück, die zeigen, dass sie benutzerfähig ist.

Die Verwendung des Attributs **-p** im Befehl **setmqaut** erteilt keinen Zugriff für alle Benutzer in derselben Primärgruppe, wenn benutzerbasierte Berechtigungen in der Datei `qm.ini` aktiviert sind, wie in der Zeilengruppe 'Service' der Datei qm.inibeschrieben.

Wenn Sie eine benutzerbasierte Berechtigung verwenden und viele Benutzer haben, werden wahrscheinlich mehr Datensätze in der AUTH-Warteschlange gespeichert als mit dem gruppenbasierten Modell, und der Berechtigungsprozess kann etwas länger dauern als vorher, da mehr Datensätze zu prüfen sind. Diese Zunahme dürfte nicht von Bedeutung sein. Falls erforderlich, können Sie eine Mischung aus Benutzer-und Gruppenberechtigungen verwenden.

Hinweise zur Migration

Wenn Sie das Modell von der Gruppe in den Benutzer eines vorhandenen Warteschlangenmanagers ändern, wird keine unmittelbare Auswirkung mehr. Die Berechtigungen, die bereits gemacht wurden, gelten weiterhin. Jeder Benutzer, der die Verbindung zum Warteschlangenmanager herstellt, erhält dieselben Berechtigungen wie zuvor: die Kombination aller Gruppen, zu denen ihre ID gehört. Wenn neue **setmqaut** -Befehle für Benutzer-IDs ausgegeben werden, werden sie sofort wirksam.

Wenn Sie einen neuen Warteschlangenmanager mit der Benutzerrichtlinie erstellen, verfügt dieser Warteschlangenmanager nur über Berechtigungen für den Benutzer, der ihn erstellt (der normalerweise die `mqm` -Benutzer-ID ist, aber nicht notwendigerweise). Es gibt auch Berechtigungen, die automatisch der Gruppe `mqm` erteilt werden. Wenn Sie jedoch nicht `mqm` als Primärgruppe haben, wird die Gruppe `mqm` nicht in die Anfangsgruppe der Berechtigungen aufgenommen.

Wenn Sie von einem Benutzer zur Gruppenrichtlinie wechseln, werden die Benutzerberechtigungen nicht automatisch gelöscht. Sie werden jedoch während der Berechtigschecks nicht mehr verwendet. Bevor Sie die Richtlinie zurücksetzen, speichern Sie die aktuelle Konfiguration, ändern Sie die Richtlinie, starten Sie den Warteschlangenmanager erneut, und wiederholen Sie anschließend das Script. Da es sich jetzt um einen gruppenbasierten Warteschlangenmanager handelt, ist der Effekt, dass die Benutzer-ID-Regeln basierend auf der Primärgruppe gespeichert werden.

Zugehörige Konzepte

Objektberechtigungsmanager (OAM)

Prinzipals und Gruppen unter UNIX, Linux und Windows

Zeilengruppe 'Service' in der Datei 'qm.ini'

Zugehörige Verweise

Befehl **crtmqm** (Warteschlangenmanager erstellen)

ALW Zugriff auf ein IBM MQ-Objekt unter AIX, Linux, and Windows erteilen

Mit dem Steuerbefehl **setmqaut**, dem MQSC-Befehl **SET AUTHREC** oder dem PCF-Befehl **MQCMD_SET_AUTH_REC** können Sie Benutzern und Benutzergruppen Zugriff auf IBM MQ-Objekte erteilen. Beachten Sie, dass Sie unter IBM MQ Appliance nur den Befehl **SET AUTHREC** verwenden können.

Eine vollständige Definition des **setmqaut** -Steuerbefehls und seiner Syntax finden Sie unter setmqaut.

Eine vollständige Definition des MQSC-Befehls **SET AUTHREC** und seiner Syntax finden Sie unter SET AUTHREC.

Eine vollständige Definition des **MQCMD_SET_AUTH_REC** -PCF-Befehls und seiner Syntax finden Sie unter Set Authority Record.

Der WS-Manager muss aktiv sein, um diesen Befehl verwenden zu können. Wenn Sie den Zugriff für einen Principal geändert haben, werden die Änderungen sofort durch den OAM widergespiegelte.

Um Benutzern Zugriff auf ein Objekt zu erteilen, müssen Sie Folgendes angeben:

- Der Name des Warteschlangenmanagers, der Eigner der Objekte ist, mit denen gearbeitet wird. Wenn Sie nicht den Namen eines Warteschlangenmanagers angeben, wird der Standardwarteschlangenmanager angenommen.
- Der Name und der Typ des Objekts (zur eindeutigen Identifizierung des Objekts). Sie geben den Namen als *Profil* an. Dies ist entweder der explizite Name des Objekts oder ein generischer Name, einschließlich Platzhalterzeichen. Eine ausführliche Beschreibung generischer Profile und der darin möglichen Platzhalterzeichen finden Sie im Abschnitt „Generische OAM-Profile unter AIX, Linux, and Windows verwenden” auf Seite 380.
- Ein oder mehrere Principals und Gruppennamen, für die die Berechtigung gilt.

  Wenn eine Benutzer-ID Leerzeichen enthält, schließen Sie sie in Anführungszeichen ein, wenn Sie diesen Befehl verwenden. Auf Windows-Systemen können Sie eine Benutzer-ID mit einem Domänennamen qualifizieren. Wenn die tatsächliche Benutzer-ID ein Zeichen (@) enthält, ersetzen Sie es durch @ @, um anzuzeigen, dass es Teil der Benutzer-ID und nicht der Begrenzer zwischen der Benutzer-ID und dem Domänennamen ist.
- Eine Liste der Berechtigungen. Jedes Element in der Liste gibt einen Zugriffstyp an, der für dieses Objekt erteilt werden soll (oder entzogen werden). Jede Berechtigung in der Liste wird als Schlüsselwort angegeben, das mit einem Pluszeichen (+) oder einem Minuszeichen (-) als Präfix versehen ist. Verwenden Sie ein Pluszeichen, um die angegebene Berechtigung hinzuzufügen, und ein Minuszeichen, um die Berechtigung zu entfernen. Zwischen dem Pluszeichen (+ oder-) und dem Schlüsselwort darf es keine Leerzeichen geben.

  Sie können eine beliebige Anzahl von Berechtigungen in einem einzigen Befehl angeben. Beispiel: Die Liste der Berechtigungen, die es einem Benutzer oder einer Gruppe ermöglichen, Nachrichten in eine Warteschlange zu stellen und sie zu durchsuchen, aber den Zugriff zum Abrufen von Nachrichten zu widerrufen, lautet:

```
+browse -get +put
```

Beispiele für die Verwendung des Befehls `setmqaut`

Die folgenden Beispiele zeigen, wie der Befehl `setmqaut` verwendet wird, um die Berechtigung zur Verwendung eines Objekts zu erteilen und zu widerrufen:

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

In diesem Beispiel gilt Folgendes:

- `saturn.queue.manager` ist der Name des Warteschlangenmanagers.
- `queue` ist der Objekttyp.
- `RED.LOCAL.QUEUE` ist der Objektname.
- `groupa` ist die ID der Gruppe mit Berechtigungen, die geändert werden sollen.
- `+browse -get +put` ist die Berechtigungsliste für die angegebene Warteschlange.
  - `+browse` fügt die Berechtigung zum Durchsuchen von Nachrichten in der Warteschlange hinzu (um **MQGET** mit der Anzeigeoption auszugeben)
  - `-get` entfernt die Berechtigung zum Abrufen (**MQGET**) von Nachrichten aus der Warteschlange
  - `+put` fügt die Berechtigung zum Einreihen (**MQPUT**) von Nachrichten in die Warteschlange hinzu.

Mit dem folgenden Befehl wird die Berechtigung put für die Warteschlange MeineWarteschlange vom Principal fvuser und von den Gruppen groupa und groupb entzogen. Auf Systemen mit AIX and Linux

wird mit diesem Befehl außerdem die Berechtigung zum Einreihen für alle Prinzipals in der gleichen Primärgruppe wie 'fvuser' entzogen.

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -p fvuser
         -g groupa -g groupb -put
```

Befehl setmqaut mit einem anderen Berechtigungsservice verwenden

Wenn Sie Ihren eigenen Berechtigungsservice anstelle des OAMs verwenden, können Sie den Namen dieses Service im Befehl **setmqaut** angeben, um den Befehl an diesen Service weiterzuleiten. Sie müssen diesen Parameter angeben, wenn mehrere installierbare Komponenten gleichzeitig ausgeführt werden. Ist dies nicht der Fall, wird die Aktualisierung an der ersten installierbaren Komponente für den Berechtigungsservice vorgenommen. Dies ist standardmäßig der bereitgestellte OAM.

Hinweise zur Verwendung von SET AUTHREC

Bei der Liste mit den Berechtigungen, die hinzugefügt werden sollen, und der Liste mit den Berechtigungen, die entfernt werden sollen, darf es keine Überschneidungen geben. Beispielsweise kann eine Anzeigeberechtigung nicht mit demselben Befehl hinzugefügt und entfernt werden. Diese Regel gilt auch dann, wenn die Berechtigungen mit verschiedenen Optionen ausgedrückt werden. Der folgende Befehl schlägt zum Beispiel fehl, weil sich die DSP-Berechtigung mit der ALLADM-Berechtigung überschneidet:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

Eine Ausnahme bei diesem Verhalten bei Überschneidungen ist die Berechtigung ALL. Mit dem folgenden Befehl werden zuerst alle ALL-Berechtigungen hinzugefügt, anschließend wird die Berechtigung SETID entfernt:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

Mit dem folgenden Befehl werden zuerst alle ALL-Berechtigungen entfernt, anschließend wird die Berechtigung DSP hinzugefügt:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

Die ALL-Berechtigungen werden immer zuerst verarbeitet, unabhängig von der Reihenfolge, in der sie im Befehl angegeben sind.

ALW Generische OAM-Profile unter AIX, Linux, and Windows verwenden

Verwenden Sie generische OAM-Profile, um in einer einzigen Operation die Berechtigungen eines Benutzers für viele Objekte festzulegen, anstatt separate **setmqaut** -Befehle oder **SET AUTHREC** -Befehle für jedes einzelne Objekt abzusetzen, wenn es erstellt wird. Beachten Sie, dass Sie unter IBM MQ Appliance nur den Befehl **SET AUTHREC** verwenden können.

Wenn Sie generische Profile in den Befehlen setmqaut oder SET AUTHREC verwenden, können Sie eine generische Berechtigung für alle Objekte festlegen, die diesem Profil entsprechen.

In dieser Sammlung von Themen wird die Verwendung generischer Profile detaillierter beschrieben.

Platzhalterzeichen in OAM-Profilen verwenden

Was ein Profil generisch macht, ist die Verwendung von Sonderzeichen (Platzhalterzeichen) im Profilnamen. Beispielsweise stimmt das Platzhalterzeichen Fragezeichen (?) mit einem beliebigen einzelnen Zeichen in einem Namen überein. Wenn Sie ABC.?EF angeben, gilt die Berechtigung, die Sie diesem Profil erteilen, für alle Objekte, die die Namen ABC.DEF, ABC.CEF, ABC.BEF usw. haben.

Folgende Platzhalterzeichen stehen zur Verfügung:

?

Verwenden Sie das Fragezeichen (?) anstelle eines beliebigen einzelnen Zeichens. `AB.?D` gilt z. B. für die Objekte `AB.CD`, `AB.ED` und `AB.FD`.

*****

Verwenden Sie den Stern (*) wie folgt:

- Ein *Qualifikationsmerkmal* in einem Profilnamen, das einem beliebigen Qualifikationsmerkmal in einem Objektnamen entspricht. Ein Qualifikationsmerkmal ist der Teil eines Objektnamens, der durch einen Punkt begrenzt wird. In `ABC.DEF.GHI`, beispielsweise sind die Qualifikationsmerkmale `ABC`, `DEF` und `GHI`.

  `ABC.*.JKL` gilt z. B. für die Objekte `ABC.DEF.JKL` und `ABC.GHI.JKL`. (Beachten Sie, dass es **nicht** für `ABC.JKL` gilt; * * verwendet in diesem Kontext immer ein Qualifikationsmerkmal.)
- Ein Zeichen in einem Qualifikationsmerkmal in einem Profilnamen, das null oder mehr Zeichen innerhalb des Qualifikationsmerkmals in einem Objektnamen entspricht.

  `ABC.DE*.JKL` gilt z. B. für die Objekte `ABC.DE.JKL`, `ABC.DEF.JKL` und `ABC.DEGH.JKL`.

******

Verwenden Sie den doppelten Stern (**) **einmal** in einem Profilnamen wie folgt:

- Der gesamte Profilname, der mit allen Objektnamen übereinstimmt. Wenn Sie beispielsweise `-t prcs` zum Identifizieren von Prozessen verwenden und ** als Profilnamen verwenden, ändern Sie die Berechtigungen für alle Prozesse.
- Als Anfangs-, Mittel-oder Endqualifikationsmerkmal in einem Profilnamen, der null oder mehr Qualifikationsmerkmale in einem Objektnamen entspricht. `**.ABC` identifiziert beispielsweise alle Objekte mit dem endgültigen Qualifikationsmerkmal ABC.

Sie können nur den doppelten Stern ** als vollständiges Qualifikationsmerkmal verwenden:

```
**.DEF
  ABC.**
  A*.**
```

aber nicht als

```
A**
```

Andernfalls erhalten Sie die Nachricht AMQ7226E: `Der Profilname ist ungültig.`

Anmerkung: Wenn Sie Platzhalterzeichen auf AIX and Linux-Systemen verwenden, **müssen** Sie den Profilnamen in einfache Anführungszeichen setzen.

Profilprioritäten

Ein wichtiger Punkt, der bei der Verwendung generischer Profile zu verstehen ist, ist die Priorität, die bei der Entscheidung, welche Berechtigungen für ein zu erstellendes Objekt angewendet werden sollen, angegeben wird. Angenommen, Sie haben die folgenden Befehle ausgegeben:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

Die erste erteilt allen Warteschlangen für den Principal fred mit Namen, die dem Profil AB. * entsprechen, die Berechtigung zum Einreihen. Der zweite Befehl erteilt die Abrufberechtigung für dieselben Warteschlangentypen, die dem Profil AB.C*.

Angenommen, Sie erstellen jetzt eine Warteschlange mit dem Namen AB.CD. Entsprechend den Regeln für die Suche nach Platzhalterzeichen kann setmqaut auf diese Warteschlange angewendet werden. Hat sie also die Befugnis erhalten oder erhalten?

Um die Antwort zu finden, wenden Sie die Regel an, die jedes Mal, wenn mehrere Profile auf ein Objekt angewendet werden können, **nur die spezifischsten gilt** . Die Art und Weise, wie Sie diese Regel anwenden, ist, indem die Profilnamen von links nach rechts verglichen werden. Unabhängig davon, wo sie sich

unterscheiden, ist ein nicht generisches Zeichen spezifischer als ein generisches Zeichen. In diesem Beispiel hat die Warteschlange AB.CD die Berechtigung **get** (AB.C* ist spezifischer als AB. *).

Wenn Sie generische Zeichen vergleichen, lautet die Reihenfolge der *Spezifität* :

1. ?
2. *
3. **

Speicherauszugsprofileinstellungen

Eine vollständige Definition des Steuerbefehls **dmpmqaut** und seiner Syntax finden Sie im Abschnitt dmpmqaut.

Eine vollständige Definition des MQSC-Befehls **DISPLAY AUTHREC** und seiner Syntax finden Sie unter DISPLAY AUTHREC..

Eine vollständige Definition des PCF-Befehls **MQCMD_INQUIRE_AUTH_RECS** und seine Syntax finden Sie unter Inquire Authority Records.

Die folgenden Beispiele zeigen die Verwendung des Steuerbefehls **dmpmqaut** zum Erstellen eines Speicherauszugs von Berechtigungssätzen für generische Profile:

1. In diesem Beispiel wird ein Speicherauszug aller Berechtigungsdatensätze mit einem Profil erstellt, das mit der Warteschlange a.b.c für den Principal user1 übereinstimmt.

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   Der resultierende Speicherauszug sieht in etwa wie folgt aus:

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

   Anmerkung: Obwohl Benutzer in AIX and Linux die Option `-p` für den Befehl **dmpmqaut** verwenden können, müssen sie stattdessen `-g groupname` verwenden, wenn Berechtigungen definiert werden.
2. In diesem Beispiel wird ein Speicherauszug aller Berechtigungsdatensätze mit einem Profil erstellt, das mit der Warteschlange a.b.c. übereinstimmt.

   ```
   dmpmqaut -m qmgr1 -n a.b.c -t q
   ```

   Der resultierende Speicherauszug sieht in etwa wie folgt aus:

   ```
   profile:     a.b.c
   object type: queue
   entity:      Administrator
   type:        principal
   authority:   all
   - - - - - - - - - - - - - - - -
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   - - - - - - - - - - - - - - - -
   profile:     a.**
   object type: queue
   entity:      group1
   type:        group
   authority:   get
   ```
3. In diesem Beispiel wird ein Speicherauszug aller Berechtigungssätze für Profil a.berstellt. *, des Typs 'Warteschlange'.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

Der resultierende Speicherauszug sieht in etwa wie folgt aus:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. In diesem Beispiel wird ein Speicherauszug aller Berechtigungsdatensätze für den Warteschlangenmanager qmX erstellt.

```
dmpmqaut -m qmX
```

Der resultierende Speicherauszug sieht in etwa wie folgt aus:

```
profile:     q1
object type: queue
entity:      Administrator
type:        principal
authority:   all
- - - - - - - - - - - - - - - -
profile:     q*
object type: queue
entity:      user1
type:        principal
authority:   get, browse
- - - - - - - - - - - - - - - -
profile:     name.*
object type: namelist
entity:      user2
type:        principal
authority:   get
- - - - - - - - - - - - - - - -
profile:     pr1
object type: process
entity:      group1
type:        group
authority:   get
```

5. In diesem Beispiel werden alle Profilnamen und Objekttypen für WS-Manager qmX erstellt.

```
dmpmqaut -m qmX -l
```

Der resultierende Speicherauszug sieht in etwa wie folgt aus:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

Anmerkung: Nur bei IBM MQ for Windows werden für alle angezeigten Principals Domäneninformationen einbezogen, zum Beispiel:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW *Platzhalterzeichen in OAM-Profilen unter AIX, Linux, and Windows verwenden*

Verwenden Sie Platzhalterzeichen in einem OAM-Profilnamen (Object Authority Manager, Objektberechtigungsmanager), um dieses Profil auf mehrere Objekte anzuwenden.

Was ein Profil generisch macht, ist die Verwendung von Sonderzeichen (Platzhalterzeichen) im Profilnamen. Beispielsweise stimmt das Platzhalterzeichen Fragezeichen (?) mit einem beliebigen einzelnen Zeichen in einem Namen überein. Wenn Sie `ABC.?EF` angeben, gilt die Berechtigung, die Sie diesem Profil erteilen, für alle Objekte, die die Namen `ABC.DEF`, `ABC.CEF`, `ABC.BEF` usw. haben.

Folgende Platzhalterzeichen stehen zur Verfügung:

?

Verwenden Sie das Fragezeichen (?) anstelle eines beliebigen einzelnen Zeichens. `AB.?D` gilt z. B. für die Objekte `AB.CD`, `AB.ED` und `AB.FD`.

*****

Verwenden Sie den Stern (*) wie folgt:

- Ein *Qualifikationsmerkmal* in einem Profilnamen, das einem beliebigen Qualifikationsmerkmal in einem Objektnamen entspricht. Ein Qualifikationsmerkmal ist der Teil eines Objektnamens, der durch einen Punkt begrenzt wird. In `ABC.DEF.GHI`, beispielsweise sind die Qualifikationsmerkmale `ABC`, `DEF` und `GHI`.

  `ABC.*.JKL` gilt z. B. für die Objekte `ABC.DEF.JKL` und `ABC.GHI.JKL`. (Beachten Sie, dass es **nicht** für `ABC.JKL` gilt; * * verwendet in diesem Kontext immer ein Qualifikationsmerkmal.)
- Ein Zeichen in einem Qualifikationsmerkmal in einem Profilnamen, das null oder mehr Zeichen innerhalb des Qualifikationsmerkmals in einem Objektnamen entspricht.

  `ABC.DE*.JKL` gilt z. B. für die Objekte `ABC.DE.JKL`, `ABC.DEF.JKL` und `ABC.DEGH.JKL`.

******

Verwenden Sie den doppelten Stern (**) **einmal** in einem Profilnamen wie folgt:

- Der gesamte Profilname, der mit allen Objektnamen übereinstimmt. Wenn Sie beispielsweise `-t prcs` zum Identifizieren von Prozessen verwenden und ** als Profilnamen verwenden, ändern Sie die Berechtigungen für alle Prozesse.
- Als Anfangs-, Mittel-oder Endqualifikationsmerkmal in einem Profilnamen, der null oder mehr Qualifikationsmerkmale in einem Objektnamen entspricht. `**.ABC` identifiziert beispielsweise alle Objekte mit dem endgültigen Qualifikationsmerkmal ABC.

Anmerkung: Wenn Sie Platzhalterzeichen auf AIX and Linux-Systemen verwenden, **müssen** Sie den Profilnamen in einfache Anführungszeichen setzen.

ALW *Profilprioritäten unter AIX, Linux, and Windows*

Mehr als ein generisches Profil kann auf ein einzelnes Objekt angewendet werden. Wo dies der Fall ist, gilt die spezifischste Regel.

Ein wichtiger Punkt, der bei der Verwendung generischer Profile zu verstehen ist, ist die Priorität, die bei der Entscheidung, welche Berechtigungen für ein zu erstellendes Objekt angewendet werden sollen, angegeben wird. Angenommen, Sie haben die folgenden Befehle ausgegeben:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

Die erste erteilt allen Warteschlangen für den Principal fred mit Namen, die dem Profil AB. * entsprechen, die Berechtigung zum Einreihen. Der zweite Befehl erteilt die Abrufberechtigung für dieselben Warteschlangentypen, die dem Profil AB.C*.

Angenommen, Sie erstellen jetzt eine Warteschlange mit dem Namen AB.CD. Entsprechend den Regeln für die Suche nach Platzhalterzeichen kann setmqaut auf diese Warteschlange angewendet werden. Hat sie also die Befugnis erhalten oder erhalten?

Um die Antwort zu finden, wenden Sie die Regel an, die jedes Mal, wenn mehrere Profile auf ein Objekt angewendet werden können, **nur die spezifischsten gilt** . Die Art und Weise, wie Sie diese Regel anwenden, ist, indem die Profilnamen von links nach rechts verglichen werden. Unabhängig davon, wo sie sich unterscheiden, ist ein nicht generisches Zeichen spezifischer als ein generisches Zeichen. In diesem Beispiel hat die Warteschlange AB.CD die Berechtigung **get** (AB.C* ist spezifischer als AB. *).

Wenn Sie generische Zeichen vergleichen, lautet die Reihenfolge der *Spezifität* :

1. ?
2. *
3. **

Informationen zur Verwendung dieses MQSC-Befehls finden Sie unter SET AUTHREC für die entsprechenden Informationen.

ALW *Speicherauszug für Profileinstellungen unter AIX, Linux, and Windows erstellen*

Mit dem Steuerbefehl **dmpmqaut** , dem MQSC-Befehl **DISPLAY AUTHREC** oder dem PCF-Befehl **MQCMD_INQUIRE_AUTH_RECS** können Sie einen Speicherauszug der aktuellen Berechtigungen erstellen, die einem angegebenen Profil zugeordnet sind. Beachten Sie, dass Sie unter IBM MQ Appliance nur den Befehl **DISPLAY AUTHREC** verwenden können.

Eine vollständige Definition des Steuerbefehls **dmpmqaut** und seiner Syntax finden Sie im Abschnitt dmpmqaut.

Eine vollständige Definition des MQSC-Befehls **DISPLAY AUTHREC** und seiner Syntax finden Sie unter DISPLAY AUTHREC..

Eine vollständige Definition des PCF-Befehls **MQCMD_INQUIRE_AUTH_RECS** und seine Syntax finden Sie unter Inquire Authority Records.

Die folgenden Beispiele zeigen die Verwendung des Steuerbefehls **dmpmqaut** zum Erstellen eines Speicherauszugs von Berechtigungssätzen für generische Profile:

1. In diesem Beispiel wird ein Speicherauszug aller Berechtigungsdatensätze mit einem Profil erstellt, das mit der Warteschlange a.b.c für den Principal user1 übereinstimmt.

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   Der resultierende Speicherauszug sieht in etwa wie in diesem Beispiel aus:

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

   Anmerkung: AIX and Linux-Benutzer können die Option `-p` nicht verwenden, sondern müssen stattdessen `-g groupname` verwenden.
2. In diesem Beispiel wird ein Speicherauszug aller Berechtigungsdatensätze mit einem Profil erstellt, das mit der Warteschlange a.b.c. übereinstimmt.

   ```
   dmpmqaut -m qmgr1 -n a.b.c -t q
   ```

   Der resultierende Speicherauszug sieht in etwa wie in diesem Beispiel aus:

   ```
   profile:     a.b.c
   object type: queue
   entity:      Administrator
   type:        principal
   authority:   all
   - - - - - - - - - - - - - - - -
   ```

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
- - - - - - - - - - - - - - - -
profile:     a.**
object type: queue
entity:      group1
type:        group
authority:   get
```

3. In diesem Beispiel wird ein Speicherauszug aller Berechtigungssätze für Profil a.berstellt. *, des Typs 'Warteschlange'.

   ```
   dmpmqaut -m qmgr1 -n a.b.* -t q
   ```

   Der resultierende Speicherauszug sieht in etwa wie in diesem Beispiel aus:

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

4. In diesem Beispiel wird ein Speicherauszug aller Berechtigungsdatensätze für den Warteschlangenmanager qmX erstellt.

   ```
   dmpmqaut -m qmX
   ```

   Der resultierende Speicherauszug sieht in etwa wie in diesem Beispiel aus:

   ```
   profile:     q1
   object type: queue
   entity:      Administrator
   type:        principal
   authority:   all
   - - - - - - - - - - - - - - - -
   profile:     q*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse
   - - - - - - - - - - - - - - - -
   profile:     name.*
   object type: namelist
   entity:      user2
   type:        principal
   authority:   get
   - - - - - - - - - - - - - - - -
   profile:     pr1
   object type: process
   entity:      group1
   type:        group
   authority:   get
   ```

5. In diesem Beispiel werden alle Profilnamen und Objekttypen für WS-Manager qmX erstellt.

   ```
   dmpmqaut -m qmX -l
   ```

   Der resultierende Speicherauszug sieht in etwa wie in diesem Beispiel aus:

   ```
   profile: q1, type: queue
   profile: q*, type: queue
   profile: name.*, type: namelist
   profile: pr1, type: process
   ```

Anmerkung: Nur bei IBM MQ for Windows werden für alle angezeigten Principals Domäneninformationen einbezogen, zum Beispiel:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW Zugriffseinstellungen unter AIX, Linux, and Windows anzeigen

Mit dem Steuerbefehl **dspmqaut** , dem MQSC-Befehl **DISPLAY AUTHREC** oder dem PCF-Befehl **MQCMD_INQUIRE_ENTITY_AUTH** können Sie die Berechtigungen anzeigen, die ein bestimmter Principal oder eine bestimmte Gruppe für ein bestimmtes Objekt hat. Beachten Sie, dass Sie unter IBM MQ Appliance nur den Befehl **DISPLAY AUTHREC** verwenden können.

Der WS-Manager muss aktiv sein, um diesen Befehl verwenden zu können. Wenn Sie den Zugriff für einen Principal ändern, werden die Änderungen sofort durch den OAM widergespiegelte. Die Berechtigung kann nur für eine Gruppe oder einen Principal gleichzeitig angezeigt werden.

Eine vollständige Definition des Steuerbefehls **dmpmqaut** und seiner Syntax finden Sie im Abschnitt dmpmqaut.

Eine vollständige Definition des MQSC-Befehls **DISPLAY AUTHREC** und seiner Syntax finden Sie unter DISPLAY AUTHREC..

Eine vollständige Definition des PCF-Befehls **MQCMD_INQUIRE_AUTH_RECS** und seine Syntax finden Sie unter Inquire Authority Records.

Das folgende Beispiel zeigt die Verwendung des Steuerbefehls **dspmqaut** zum Anzeigen der Berechtigungen, die die Gruppe `GpAdmin` für eine Prozessdefinition mit dem Namen `Annuities` im Warteschlangenmanager `QueueMan1`hat.

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

ALW Zugriff auf ein IBM MQ-Objekt unter AIX, Linux, and Windows ändern und widerrufen

Verwenden Sie den Steuerbefehl **setmqaut** , den MQSC-Befehl **DELETE AUTHREC** oder den PCF-Befehl **MQCMD_DELETE_AUTH_REC** , um die Zugriffsebene zu ändern, die ein Benutzer oder eine Gruppe auf ein Objekt hat. MQ Appliance Beachten Sie, dass unter IBM MQ Appliance nur der Befehl **DELETE AUTHREC** verwendet werden kann.

Der Prozess, mit dem der Benutzer aus einer Gruppe entfernt wird, wird in beschrieben:

- Windows „Gruppen unter Windows erstellen und verwalten” auf Seite 158
- AIX „Gruppen unter AIX erstellen und verwalten” auf Seite 156
- Linux „Gruppen unter Linux erstellen und verwalten” auf Seite 157

.

Die Benutzer-ID, mit der ein IBM MQ-Objekt erstellt wird, erhält uneingeschränkte Zugriffsberechtigungen für dieses Objekt. Wenn Sie diese Benutzer-ID aus der lokalen Gruppe 'mqm' entfernen (oder auf Windows-Systemen aus der Administratorgruppe), werden diese Berechtigungen nicht widerrufen. Verwenden Sie den Steuerbefehl **setmqaut** oder den PCF-Befehl **MQCMD_DELETE_AUTH_REC** , um den Zugriff auf ein Objekt für die Benutzer-ID, die es erstellt hat, zu widerrufen, nachdem es aus der Gruppe 'mqm' oder 'Administratoren' entfernt wurde.

Eine vollständige Definition des Befehls "setmqaut control" und seiner Syntax finden Sie in setmqaut .

Eine vollständige Definition des MQSC-Befehls **DELETE AUTHREC** und seiner Syntax finden Sie unter DELETE AUTHREC.

Eine vollständige Definition des PCF-Befehls **MQCMD_DELETE_AUTH_REC** und dessen Syntax finden Sie unter Berechtigungssatz löschen.

Windows Unter Windows können Sie ab IBM MQ 8.0 die OAM-Einträge für ein bestimmtes Windows-Benutzerkonto jederzeit mit dem Parameter **-u** *SID* für den Befehl **setmqaut** entfernen.

Vor IBM MQ 8.0 mussten Sie die OAM-Einträge für ein bestimmtes Windows-Benutzerkonto vor dem Löschen des Benutzerprofils entfernen. Es war nicht möglich, die OAM-Einträge nach dem Entfernen des Benutzerkontos zu entfernen.

ALW Sicherheitszugriffsprüfungen auf Systemen mit AIX, Linux, and Windows verhindern

Hinweis: In diesem Abschnitt werden Funktionen beschrieben, deren Aktivierung nicht empfohlen wird. Zum Inaktivieren der Sicherheitsprüfung können Sie den Objektberechtigungsmanager (Object Authority Manager, OAM) inaktivieren. Dies kann für eine Testumgebung geeignet sein. Wenn diese Option inaktiviert ist, kann der Warteschlangenmanager keine Berechtigungs-oder Verbindungsauthentifizierungsprüfungen mehr durchführen. TLS, Kanalauthentifizierungsdatensätze und Sicherheitsexits können weiterhin verwendet werden. Wenn Sie den OAM inaktiviert oder entfernt haben, können Sie keinen OAM einem vorhandenen WS-Manager hinzufügen.

Wenn Sie nicht möchten, dass Sicherheitsprüfungen (z. B. in einer Testumgebung) ausgeführt werden, können Sie den OAM auf eine der folgenden Arten inaktivieren:

- Bevor Sie einen Warteschlangenmanager erstellen, müssen Sie die Betriebssystemumgebungsvariable MQSNOAUT festlegen.

  Informationen zu den Auswirkungen der Einstellung der Variablen MQSNOAUT und zur Festlegung von MQSNOAUT unter AIX, Linux, and Windowsfinden Sie im Abschnitt Beschreibungen der Umgebungsvariablen.
- Bearbeiten Sie die Konfigurationsdatei des Warteschlangenmanagers, um den Service zu entfernen.

Warnung: Wenn ein OAM entfernt wird, kann er nicht auf einen vorhandenen Warteschlangenmanager zurückgestellt werden. Dies liegt daran, dass der OAM zur Objekterstellungszeit vorhanden sein muss. Wenn Sie den IBM MQ-OAM nach dem Löschen erneut verwenden möchten, müssen Sie den Warteschlangenmanager erneut erstellen.

Wenn Sie den Befehl **setmqaut**oder **dspmqaut** verwenden, während der OAM inaktiviert ist, beachten Sie die folgenden Punkte:

- Der OAM prüft den angegebenen Principal oder die angegebene Gruppe nicht. Dies bedeutet, dass der Befehl ungültige Werte akzeptieren kann.
- Der OAM führt keine Sicherheitsprüfungen durch und zeigt an, dass alle Principals und Gruppen berechtigt sind, alle anwendbaren Objektoperationen auszuführen.
- Alle Berechtigungsnachweise, die für Authentifizierungsprüfungen an den OAM übergeben werden, werden nicht validiert.

Zugehörige Tasks

Installierbare Services konfigurieren

Zugehörige Verweise

Installierbare Services und Komponenten für UNIX, Linux und Windows

Referenzinformationen zu installierbaren Services

Erforderlicher Zugriff auf Ressourcen erteilen

Verwenden Sie dieses Topic, um festzustellen, welche Tasks ausgeführt werden sollen, um die Sicherheit Ihres IBM MQ-Systems zu erhöhen.

Informationen zu diesem Vorgang

Während dieser Task legen Sie fest, welche Aktionen erforderlich sind, um die entsprechende Ebene der Sicherheit für die Elemente Ihrer IBM MQ-Installation anzuwenden. Jede einzelne Aufgabe, auf die Sie Bezug genommen haben, enthält Schritt-by-Schritt-Anleitungen für alle Plattformen.

Vorgehensweise

1. Müssen Sie den Zugriff auf den WS-Manager auf bestimmte Benutzer beschränken?
   a) Nein: Nehmen Sie keine weitere Aktion vor.
   b) Ja: Fahren Sie mit der nächsten Frage fort.
2. Benötigen diese Benutzer einen partiellen Verwaltungszugriff auf eine Untergruppe von Warteschlangenmanagerressourcen?
   a) Nein: Fahren Sie mit der nächsten Frage fort.
   b) Ja: Siehe „Teilweiser Verwaltungszugriff auf eine Untergruppe von Warteschlangenmanagerressourcen erteilen” auf Seite 389.
3. Benötigen diese Benutzer uneingeschränkten Verwaltungszugriff auf eine Untergruppe von Ressourcen des Warteschlangenmanagers?
   a) Nein: Fahren Sie mit der nächsten Frage fort.
   b) Ja: Siehe „Vollzugriff auf Verwaltungszugriff auf eine Untergruppe von Warteschlangenmanagerressourcen erteilen” auf Seite 399.
4. Benötigen diese Benutzer nur Lesezugriff auf alle WS-Manager-Ressourcen?
   a) Nein: Fahren Sie mit der nächsten Frage fort.
   b) Ja: Siehe „Schreibgeschützter Zugriff auf alle Ressourcen in einem Warteschlangenmanager erteilen” auf Seite 405.
5. Benötigen diese Benutzer uneingeschränkten Verwaltungszugriff auf alle WS-Manager-Ressourcen?
   a) Nein: Fahren Sie mit der nächsten Frage fort.
   b) Ja: Siehe „Vollzugriff Verwaltungszugriff auf alle Ressourcen in einem WS-Manager erteilen” auf Seite 407.
6. Benötigen Sie Benutzeranwendungen, um eine Verbindung zu Ihrem Warteschlangenmanager herzustellen?
   a) Nein: Inaktivieren Sie die Verbindung, wie unter „Verbindung zum WS-Manager wird entfernt” auf Seite 408 beschrieben.
   b) Ja: Siehe „Benutzeranwendungen die Verbindung zum Warteschlangenmanager ermöglichen” auf Seite 409.

Multi z/OS Teilweiser Verwaltungszugriff auf eine Untergruppe von Warteschlangenmanagerressourcen erteilen

Sie müssen bestimmten Benutzern einen partiellen Verwaltungszugriff auf einige, aber nicht alle Warteschlangenmanagerressourcen erteilen. Verwenden Sie diese Tabelle, um die Aktionen zu ermitteln, die Sie ausführen müssen.

Tabelle 69. Teilweiser Verwaltungszugriff auf eine Untergruppe von WS-Manager-Ressourcen erteilen

Die Benutzer müssen Objekte dieses Typs verwalten.	**Diese Aktion ausführen**
Warteschlangen	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Warteschlangen, wie im Abschnitt „beschränkten Verwaltungszugriff auf einige Warteschlangen erteilen” auf Seite 390 erläutert.

Tabelle 69. Teilweiser Verwaltungszugriff auf eine Untergruppe von WS-Manager-Ressourcen erteilen (Forts.)	
Die Benutzer müssen Objekte dieses Typs verwalten.	**Diese Aktion ausführen**
Themen	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Themen, wie im Abschnitt „Erteilen eines eingeschränkten Verwaltungszugriffs auf bestimmte Themen" auf Seite 392 erläutert.
Kanäle	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Kanäle, wie im Abschnitt „beschränkten Verwaltungszugriff auf einige Kanäle erteilen" auf Seite 393 erläutert.
Der Warteschlangenmanager	Erteilen Sie partiellen Verwaltungszugriff auf den Warteschlangenmanager, wie im Abschnitt „Erteilen des eingeschränkten Verwaltungszugriffs auf einen Warteschlangenmanager" auf Seite 394 erläutert.
Prozesse	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Prozesse, wie im Abschnitt „Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Prozesse" auf Seite 395 erläutert.
Namenslisten	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Namenslisten, wie im Abschnitt „Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Namenslisten" auf Seite 396 erläutert.
Services	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Services, wie im Abschnitt „Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Services" auf Seite 398 erläutert.

beschränkten Verwaltungszugriff auf einige Warteschlangen erteilen

Erteilen Sie partiellen Verwaltungszugriff auf einige Warteschlangen in einem Warteschlangenmanager für jede Gruppe von Benutzern, die einen Geschäftsbedarf für sie haben.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um begrenzten Verwaltungszugriff auf einige Warteschlangen für einige Aktionen zu erteilen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
  ```

- IBM i
  Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS Geben Sie für z/OS die folgenden Befehle aus, um Zugriff auf eine angegebene Warteschlange zu erteilen:

```
RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Geben Sie die folgenden Befehle für jeden MQSC-Befehl an, um anzugeben, welche MQSC-Befehle der Benutzer in der Warteschlange ausführen kann:

```
RDEFINE MQCMDS QMgrName. ReqdAction. QType UACC(NONE)
PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Um dem Benutzer die Verwendung des Befehls DISPLAY QUEUE zu ermöglichen, geben Sie die folgenden Befehle aus:

```
RDEFINE MQCMDS QMgrName.DISPLAY. QType UACC(NONE)
PERMIT QMgrName.DISPLAY. QType CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers.

z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

ObjectProfile
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

GroupName
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

ReqdAction
Die Aktion, die die Gruppe ausführen kann:

- ALW Auf Systemen mit AIX, Linux, and Windows können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: +chg, +clr, +dlt, +dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.
- IBM i Unter IBM i können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: *ADMCHG, *ADMCLR, *ADMDLT, *ADMDSP. Die Berechtigung *ALLADM ist äquivalent zu allen diesen Einzelberechtigungen.
- z/OS Unter z/OS können Sie einen der Werte ALTER, CLEAR, DELETE oder MOVE verwenden.

Anmerkung: Das Erteilen von + crt für Warteschlangen macht den Benutzer oder die Gruppe indirekt zu einem Administrator. Verwenden Sie nicht die Berechtigung + crt, um begrenzten Verwaltungszugriff auf einige Warteschlangen zu erteilen.

QType

Für den Befehl DISPLAY eine der folgenden Werte: QUEUE, QLOCAL, QALIAS, QMODEL, QREMOTE oder QCLUSTER.

Für andere Werte von *ReqdAction* ist einer der Werte QLOCAL, QALIAS, QMODEL oder QREMOTE.

Erteilen eines eingeschränkten Verwaltungszugriffs auf bestimmte Themen

Erteilen Sie partiellen Verwaltungszugriff auf einige Themen in einem Warteschlangenmanager und jede Gruppe von Benutzern, die einen Geschäftsbedarf für sie haben.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um begrenzten Verwaltungszugriff auf einige Themen für einige Aktionen zu erteilen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW

  Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
  ```

- IBM i

  Geben Sie für IBM i den folgenden Befehl aus:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Geben Sie für z/OS die folgenden Befehle aus:

  ```
  RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
  PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Mit diesen Befehlen wird der Zugriff auf das angegebene Thema erteilt. Um festzustellen, welche MQSC-Befehle der Benutzer zu dem Thema ausführen kann, geben Sie die folgenden Befehle für jeden MQSC-Befehl aus:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.TOPIC UACC(NONE)
  PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Um dem Benutzer die Verwendung des Befehls DISPLAY TOPIC zu ermöglichen, geben Sie die folgenden Befehle aus:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.TOPIC UACC(NONE)
  PERMIT QMgrName.DISPLAY.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  ObjectProfile
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  GroupName
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

ReqdAction
Die Aktion, die die Gruppe ausführen kann:

- ALW Auf AIX, Linux, and Windows-Systemen jede Kombination der folgenden Berechtigungen: + chg, + clr, + crt, + dlt, + dsp. + ctrl. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.
- IBM i Unter IBM i können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL. Die Berechtigung *ALLADM ist äquivalent zu allen diesen Einzelberechtigungen.
- z/OS Unter z/OS können Sie einen der Werte ALTER, CLEAR, DEFINE, DELETE oder MOVE verwenden.

beschränkten Verwaltungszugriff auf einige Kanäle erteilen

Erteilen Sie einem Teil des Verwaltungszugriffs auf einige Kanäle in einem Warteschlangenmanager jede Gruppe von Benutzern, die einen Geschäftsbedarf für sie haben.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um begrenzten Verwaltungszugriff auf einige Kanäle für einige Aktionen zu erteilen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- IBM i
  Unter IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS Unter z/OS:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Mit diesen Befehlen wird der Zugriff auf den angegebenen Kanal erteilt. Geben Sie die folgenden Befehle für jeden MQSC-Befehl aus, um festzustellen, welche MQSC-Befehle der Benutzer auf dem Kanal ausführen kann:

```
RDEFINE MQCMDS QMgrName. ReqdAction.CHANNEL UACC(NONE)
PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Um dem Benutzer die Verwendung des Befehls DISPLAY CHANNEL zu ermöglichen, geben Sie die folgenden Befehle aus:

```
RDEFINE MQCMDS QMgrName.DISPLAY.CHANNEL UACC(NONE)
PERMIT QMgrName.DISPLAY.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers.

z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

ObjectProfile
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

GroupName
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

ReqdAction
Die Aktion, die die Gruppe ausführen kann:

- ALW Unter AIX, Linux, and Windows eine beliebige Kombination der folgenden Berechtigungen: + chg, + clr, + crt, + dlt, + dsp. + ctrl, + ctrlx. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.
- IBM i Unter IBM i können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLx. Die Berechtigung *ALLADM ist äquivalent zu allen diesen Einzelberechtigungen.
- z/OS Unter z/OS können Sie einen der Werte ALTER, CLEAR, DEFINE, DELETE oder MOVE verwenden.

Erteilen des eingeschränkten Verwaltungszugriffs auf einen Warteschlangenmanager

Erteilen Sie einem WS-Manager einen partiellen Verwaltungszugriff auf jede Gruppe von Benutzern, die einen Geschäftsbedarf für sie haben.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um begrenzten Verwaltungszugriff zu erteilen, um bestimmte Aktionen für den Warteschlangenmanager auszuführen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
```

- IBM i
  Unter IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS
  Unter z/OS:

  Geben Sie für jeden MQSC-Befehl die folgenden Befehle aus, um festzustellen, welche MQSC-Befehle Sie auf dem Warteschlangenmanager ausführen können:

```
RDEFINE MQCMDS QMgrName. ReqdAction.QMGR UACC(NONE)
PERMIT QMgrName. ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Um dem Benutzer die Verwendung des Befehls DISPLAY QMGR zu ermöglichen, geben Sie die folgenden Befehle aus:

```
RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers.

ObjectProfile
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

GroupName
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

ReqdAction
Die Aktion, die die Gruppe ausführen kann:

- ALW Unter AIX, Linux, and Windows können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: +chg, +clr, +crt, +dlt, +dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

  Obwohl + festgelegt ist eine MQI-Berechtigung, die normalerweise nicht als administrativ betrachtet wird, kann die Erteilungs-und Erteilungs-ID auf dem WS-Manager indirekt zu einer vollständigen Administratorberechtigung führen. Erteilen Sie den gewöhnlichen Benutzern und Anwendungen keine +-Gruppe.
- IBM i Unter IBM i können Sie eine Kombination aus den folgenden Berechtigungen verwenden: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. Die Berechtigung *ALLADM ist äquivalent zu allen diesen Einzelberechtigungen.

Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Prozesse

Erteilen Sie partiellen Verwaltungszugriff auf einige Prozesse in einem Warteschlangenmanager und jede Gruppe von Benutzern mit einem Geschäftsbedarf für diese Gruppe.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um begrenzten Verwaltungszugriff auf einige Prozesse für einige Aktionen zu erteilen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
  ```
- IBM i
  Unter IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS Unter z/OS:

```
RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Mit diesen Befehlen wird der Zugriff auf den angegebenen Kanal erteilt. Geben Sie die folgenden Befehle für jeden MQSC-Befehl aus, um festzustellen, welche MQSC-Befehle der Benutzer auf dem Kanal ausführen kann:

```
RDEFINE MQCMDS QMgrName. ReqdAction.PROCESS UACC(NONE)
PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Um dem Benutzer die Verwendung des Befehls DISPLAY PROCESS zu ermöglichen, geben Sie die folgenden Befehle aus:

```
RDEFINE MQCMDS QMgrName.DISPLAY.PROCESS UACC(NONE)
PERMIT QMgrName.DISPLAY.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers.

z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

ObjectProfile
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

GroupName
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

ReqdAction
Die Aktion, die die Gruppe ausführen kann:

- ALW Unter AIX, Linux, and Windows können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: +chg, +clr, +crt, +dlt, +dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.
- IBM i Unter IBM i können Sie eine Kombination aus den folgenden Berechtigungen verwenden: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. Die Berechtigung *ALLADM ist äquivalent zu allen diesen Einzelberechtigungen.
- z/OS Unter z/OS können Sie einen der Werte ALTER, CLEAR, DEFINE, DELETE oder MOVE verwenden.

Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Namenslisten

Erteilen Sie einem Teil des Verwaltungszugriffs auf einige Namenslisten in einem Warteschlangenmanager Zugriff auf jede Gruppe von Benutzern mit einem Geschäftsbedarf für diese Gruppe.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um begrenzten Verwaltungszugriff auf einige Namenslisten für einige Aktionen zu erteilen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- IBM i
  Unter IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS Unter z/OS:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Mit diesen Befehlen wird der Zugriff auf die angegebene Namensliste erteilt. Um festzustellen, welche MQSC-Befehle der Benutzer in der Namensliste ausführen kann, geben Sie die folgenden Befehle für jeden MQSC-Befehl aus:

```
RDEFINE MQCMDS QMgrName. ReqdAction.NAMELIST UACC(NONE)
PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Um dem Benutzer die Verwendung des Befehls DISPLAY NAMELIST zu ermöglichen, geben Sie die folgenden Befehle aus:

```
RDEFINE MQCMDS QMgrName.DISPLAY.NAMELIST UACC(NONE)
PERMIT QMgrName.DISPLAY.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  ObjectProfile
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  GroupName
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

  ReqdAction
  Die Aktion, die die Gruppe ausführen kann:

  – ALW Unter AIX, Linux, and Windows können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

  – IBM i Unter IBM i können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLx. Die Berechtigung *ALLADM ist äquivalent zu allen diesen Einzelberechtigungen.

– z/OS Unter z/OS können Sie einen der Werte ALTER, CLEAR, DEFINE, DELETE oder MOVE verwenden.

Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Services

Erteilen Sie einem Teil des Verwaltungszugriffs auf einige Services in einem Warteschlangenmanager jede Gruppe von Benutzern mit einem Geschäftsbedarf für diese.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um begrenzten Verwaltungszugriff auf einige Services für bestimmte Aktionen zu erteilen. z/OS Beachten Sie, dass Serviceobjekte auf z/OS nicht vorhanden sind.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW

  Unter AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
  ```

- Unter IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Unter z/OS:

  Mit diesen Befehlen wird der Zugriff auf den angegebenen Service gewährt. Um festzustellen, welche MQSC-Befehle der Benutzer für den Service ausführen kann, geben Sie die folgenden Befehle für jeden MQSC-Befehl aus:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.SERVICE UACC(NONE)
  PERMIT QMgrName. ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Um dem Benutzer die Verwendung des Befehls DISPLAY SERVICE zu ermöglichen, geben Sie die folgenden Befehle aus:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
  PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  ObjectProfile
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  GroupName
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

  ReqdAction
  Die Aktion, die die Gruppe ausführen kann:

– ALW Auf Systemen mit AIX, Linux, and Windows eine können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

– IBM i Unter IBM i können Sie eine beliebige Kombination aus den folgenden Berechtigungen verwenden: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLx. Die Berechtigung *ALLADM ist äquivalent zu allen diesen Einzelberechtigungen.

Vollzugriff auf Verwaltungszugriff auf eine Untergruppe von Warteschlangenmanagerressourcen erteilen

Sie müssen bestimmten Benutzern vollständigen Verwaltungszugriff auf einige, aber nicht alle Warteschlangenmanagerressourcen erteilen. Verwenden Sie diese Tabellen, um die Aktionen zu ermitteln, die Sie ausführen müssen.

Tabelle 70. Vollzugriff auf Verwaltungszugriff auf eine Untergruppe von Warteschlangenmanagerressourcen erteilen

Die Benutzer müssen Objekte dieses Typs verwalten.	**Diese Aktion ausführen**
Warteschlangen	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Warteschlangen, wie im Abschnitt „Vollzugriff Verwaltungszugriff auf einige Warteschlangen erteilen” auf Seite 399 erläutert.
Themen	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Themen, wie im Abschnitt „Vollenden Verwaltungszugriff auf einige Themen erteilen” auf Seite 400 erläutert.
Kanäle	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Kanäle, wie im Abschnitt „Vollenden Verwaltungszugriff auf einige Kanäle erteilen” auf Seite 401 erläutert.
Der Warteschlangenmanager	Erteilen Sie vollständigen Verwaltungszugriff auf den Warteschlangenmanager, wie im Abschnitt „Vollzugriff auf den Verwaltungszugriff auf einen Warteschlangenmanager erteilen” auf Seite 402 erläutert.
Prozesse	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Prozesse, wie im Abschnitt „Vollenden Verwaltungszugriff auf einige Prozesse erteilen” auf Seite 403 erläutert.
Namenslisten	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Namenslisten, wie im Abschnitt „Vollenden Verwaltungszugriff auf einige Namenslisten erteilen” auf Seite 404 erläutert.
Services	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Services, wie im Abschnitt „Vollenden Verwaltungszugriff auf einige Services erteilen” auf Seite 404 erläutert.

Vollzugriff Verwaltungszugriff auf einige Warteschlangen erteilen

Erteilen Sie jedem Benutzer mit einem Geschäftsbedarf für jede Gruppe von Benutzern vollständigen Verwaltungszugriff auf einige Warteschlangen in einem Warteschlangenmanager.

Informationen zu diesem Vorgang

Wenn Sie vollständigen Verwaltungszugriff auf einige Warteschlangen erteilen möchten, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW

  Unter AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
  ```

- IBM i

  Unter IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrName
  ')
  ```

- z/OS

  Unter z/OS:

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  Objekt-profil
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  Gruppenname
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Vollenden Verwaltungszugriff auf einige Themen erteilen

Erteilen Sie jedem Benutzer mit einem Geschäftsbedarf für jede Gruppe von Benutzern vollständigen Verwaltungszugriff auf einige Themen in einem Warteschlangenmanager.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um vollständigen Verwaltungszugriff auf einige Themen für einige Aktionen zu erteilen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
```

- IBM i
  Unter IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  Unter z/OS:

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  Objekt-profil
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  Gruppenname
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Vollenden Verwaltungszugriff auf einige Kanäle erteilen

Erteilen Sie jedem Benutzer mit einem Geschäftsbedarf für jede Gruppe von Benutzern vollständigen Verwaltungszugriff auf einige Kanäle in einem Warteschlangenmanager.

Informationen zu diesem Vorgang

Wenn Sie vollständigen Verwaltungszugriff auf einige Kanäle erteilen möchten, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
```

- IBM i
  Unter IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME(' QMgrNa
me ')
```

- z/OS

  Unter z/OS:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  Objekt-profil
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  Gruppenname
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Vollzugriff auf den Verwaltungszugriff auf einen Warteschlangenmanager erteilen

Erteilen Sie jedem Benutzer mit einem Geschäftsbedarf für jede Gruppe von Benutzern den vollständigen Verwaltungszugriff auf einen Warteschlangenmanager.

Informationen zu diesem Vorgang

Um vollständigen Verwaltungszugriff auf den Warteschlangenmanager zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW

  Unter AIX, Linux, and Windows:

```
setmqaut -m QMgrName -t qmgr -g GroupName +alladm
```

- IBM i

  Unter IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrNa
me ')
```

- z/OS

  Unter z/OS:

```
RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers.

z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Vollenden Verwaltungszugriff auf einige Prozesse erteilen

Erteilen Sie jedem Benutzer mit einem Geschäftsbedarf für jede Gruppe von Benutzern den vollständigen Verwaltungszugriff auf einige Prozesse auf einem Warteschlangenmanager.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um vollständigen Verwaltungszugriff auf einige Prozesse zu erteilen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
```

- IBM i
  Unter IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrNa
me ')
```

- z/OS
  Unter z/OS:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  Objekt-profil
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  Gruppenname
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Vollenden Verwaltungszugriff auf einige Namenslisten erteilen

Erteilen Sie jedem Benutzer mit einem Geschäftsbedarf für jede Gruppe von Benutzern uneingeschränkten Verwaltungszugriff auf einige Namenslisten.

Informationen zu diesem Vorgang

Um vollständigen Verwaltungszugriff auf einige Namenslisten zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW

  Unter AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
  ```

- IBM i

  Unter IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  Unter z/OS:

  ```
  RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
  PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  Objekt-profil
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  Gruppenname
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Vollenden Verwaltungszugriff auf einige Services erteilen

Erteilen Sie jedem Benutzer mit einem Geschäftsbedarf für jede Gruppe von Benutzern vollständigen Verwaltungszugriff auf einige Services auf einem Warteschlangenmanager.

Informationen zu diesem Vorgang

Verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem, um vollständigen Verwaltungszugriff auf einige Services zu erteilen.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
  ```

- IBM i
  Unter IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrNa
  me ')
  ```

- z/OS
  Unter z/OS:

  ```
  RDEFINE MQADMIN QMgrName.SERVICE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.SERVICE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers.

  z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  ObjectProfile
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  GroupName
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Schreibgeschützter Zugriff auf alle Ressourcen in einem Warteschlangenmanager erteilen

Erteilen Sie jedem Benutzer oder einer Gruppe von Benutzern mit einem Geschäftsbedarf einen schreibgeschützten Zugriff auf alle Ressourcen in einem Warteschlangenmanager.

Informationen zu diesem Vorgang

Verwenden Sie den Assistenten "Aufgabenbereichsbasierte Berechtigungen hinzufügen" oder die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Nachdem Sie Autorisierungsdetails geändert haben, führen Sie eine Sicherheitsaktualisierung mit dem Befehl SICHERHEIT AKTUALISIEREN durch.

Prozedur

- Mit dem Assistenten:

a) Klicken Sie im Navigatorfenster von IBM MQ Explorer mit der rechten Maustaste auf den Warteschlangenmanager, und klicken Sie dann auf **Objektberechtigungen** > **Rollenbasierte Berechtigungen hinzufügen**.

Der Assistent 'Rollenbasierte Berechtigungen hinzufügen' wird geöffnet.

- ALW

Geben Sie auf Systemen mit AIX, Linux, and Windows die folgenden Befehle aus:

```
setmqaut -m QMgrName -n ** -t queue -g GroupName +browse +dsp
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get
+put
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp
setmqaut -m QMgrName -n ** -t clntconn -g GroupName +dsp
setmqaut -m QMgrName -n ** -t authinfo -g GroupName +dsp
setmqaut -m QMgrName -n ** -t listener -g GroupName +dsp
setmqaut -m QMgrName -n ** -t namelist -g GroupName +dsp
setmqaut -m QMgrName -n ** -t process -g GroupName +dsp
setmqaut -m QMgrName -n ** -t service -g GroupName +dsp
setmqaut -m QMgrName -t qmgr -g GroupName +dsp +inq +connect
```

Die spezifischen Berechtigungen für SYSTEM.ADMIN.COMMAND.QUEUE und SYSTEM.MQEXPLORER.REPLY.MODEL sind nur erforderlich, wenn Sie die IBM MQ Explorerverwenden wollen.

- IBM i

Geben Sie für IBM i die folgenden Befehle aus:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*Q) USER('GroupName') AUT(*ADMDSP *BROWSE) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*TOPIC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CHL) USER('GroupName') AUT(*ADMDSP *INQ) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CLTCN) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*AUTHINFO) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*LSR) USER('GroupName') AUT(*ADMDSP)MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*NMLIST) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*SVC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ('object-name') OBJTYPE(*MQM) USER('GroupName') AUT(*ADMDSP *CONNECT *INQ)
MQMNAME('QMgrName')
```

- z/OS

Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQQUEUE QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQQUEUE) ID(GroupName) ACCESS(READ)
RDEFINE MXTOPIC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MXTOPIC) ID(GroupName) ACCESS(READ)
RDEFINE MQPROC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQPROC) ID(GroupName) ACCESS(READ)
RDEFINE MQNLIST QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers.

z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

GroupName
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Vollzugriff Verwaltungszugriff auf alle Ressourcen in einem WS-Manager erteilen

Erteilen Sie jedem Benutzer oder jeder Gruppe von Benutzern, die einen Geschäftsbedarf haben, vollständigen Verwaltungszugriff auf alle Ressourcen eines Warteschlangenmanagers.

Informationen zu diesem Vorgang

Sie können den Assistenten "Rollenbasierte Berechtigungen hinzufügen" oder die entsprechenden Befehle für Ihr Betriebssystem verwenden.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Anmerkungen: ALW

1. Wenn Sie **runmqsc** verwenden, um den Warteschlangenmanager anstelle von IBM MQ Explorerzu verwalten, müssen Sie die Berechtigung zum Abrufen, Abfragen und Durchsuchen des SYSTEM.MQSC.REPLY.QUEUEund Sie müssen keine Berechtigungen für SYSTEM.MQEXPLORER.REPLY.MODEL -Warteschlange.
2. Wenn einem Benutzer Zugriff auf alle Ressourcen auf einem Warteschlangenmanager erteilt wird, gibt es einige Befehle, die der Benutzer nicht ausführen kann, es sei denn, dieser Benutzer hat Lesezugriff auf die Datei `qm.ini` . Dies ist darauf zurückzuführen, dass Benutzer, die keine`mqm` sind, die`qm.ini` Datei nicht lesen können.

   Der Benutzer kann die folgenden Befehle nur ausführen, wenn Sie ihm Lesezugriff auf die Datei qm.ini gewährt haben:

   - Definieren eines Kanals, der für die Verwendung von TLS konfiguriert ist
   - Definieren eines Kanals mit Hilfe von Einfügevariablen zur Autokonfiguration, die in`qm.ini` definiert sind

Prozedur

- Wenn Sie den Assistenten verwenden, klicken Sie im Teilfenster IBM MQ Explorer Navigator mit der rechten Maustaste auf den Warteschlangenmanager und klicken Sie auf **Objektberechtigungen** > **Rollenbasierte Berechtigungen hinzufügen**.

  Der Assistent 'Rollenbasierte Berechtigungen hinzufügen' wird geöffnet.
- Linux AIX

  Geben Sie auf Systemen mit AIX and Linux die folgenden Befehle aus:

```
setmqaut -m QMgrName -n '**' -t queue -g GroupName +alladm +browse
setmqaut -m QMgrName -n @class -t queue -g GroupName +crt
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get +put
setmqaut -m QMgrName -n '**' -t topic -g GroupName +alladm
setmqaut -m QMgrName -n @class -t topic -g GroupName +crt
setmqaut -m QMgrName -n '**' -t channel -g GroupName +alladm
setmqaut -m QMgrName -n @class -t channel -g GroupName +crt
setmqaut -m QMgrName -n '**' -t clntconn -g GroupName +alladm
setmqaut -m QMgrName -n @class -t clntconn -g GroupName +crt
setmqaut -m QMgrName -n '**' -t authinfo -g GroupName +alladm
setmqaut -m QMgrName -n @class -t authinfo -g GroupName +crt
setmqaut -m QMgrName -n '**' -t listener -g GroupName +alladm
setmqaut -m QMgrName -n @class -t listener -g GroupName +crt
setmqaut -m QMgrName -n '**' -t namelist -g GroupName +alladm
setmqaut -m QMgrName -n @class -t namelist -g GroupName +crt
setmqaut -m QMgrName -n '**' -t process -g GroupName +alladm
setmqaut -m QMgrName -n @class -t process -g GroupName +crt
setmqaut -m QMgrName -n '**' -t service -g GroupName +alladm
```

```
setmqaut -m QMgrName -n @class -t service -g GroupName +crt
setmqaut -m QMgrName -t qmgr -g GroupName +alladm +connect
```

Weitere Informationen zu `@class` finden Sie unter **setmqaut** .

- Windows

  Geben Sie für Windows-Systeme die gleichen Befehle wie für AIX and Linux-Systeme ein, verwenden Sie anstelle von @class aber den Profilnamen @CLASS.

- IBM i

  Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER(' GroupName ') AUT(*ALLADM) MQMNAME(' QMgrName ')
```

- z/OS

  Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers.

z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

GroupName
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Verbindung zum WS-Manager wird entfernt

Wenn keine Benutzeranwendungen eine Verbindung zu Ihrem Warteschlangenmanager herstellen sollen, entfernen Sie die entsprechende Berechtigung, um eine Verbindung zu diesem Warteschlangenmanager herzustellen.

Informationen zu diesem Vorgang

Rufen Sie die Berechtigung aller Benutzer auf, eine Verbindung zum Warteschlangenmanager herzustellen, indem Sie den entsprechenden Befehl für Ihr Betriebssystem verwenden.

Unter Multiplatforms können Sie auch den Befehl DELETE AUTHREC verwenden.

Anmerkung: Unter IBM MQ Appliance können Sie nur den Befehl **DELETE AUTHREC** verwenden.

Prozedur

- ALW

  Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -t qmgr -g GroupName -connect
```

- IBM i

  Geben Sie für IBM i den folgenden Befehl aus:

```
RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
```

- z/OS

  Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
```

Geben Sie keine PERMIT-Befehle aus.

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers.

z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

GroupName
Der Name der Gruppe, der der Zugriff verweigert werden soll.

Benutzeranwendungen die Verbindung zum Warteschlangenmanager ermöglichen

Sie möchten es einer Benutzeranwendung ermöglichen, eine Verbindung zu Ihrem Warteschlangenmanager herzustellen. Anhand der Tabellen in diesem Abschnitt können Sie feststellen, welche Schritte dazu erforderlich sind.

Stellen Sie zunächst fest, ob Clientanwendungen eine Verbindung zu Ihrem Queue Manager herstellen.

Befindet sich unter den Anwendungen, die eine Verbindung zum Warteschlangenmanager herstellen sollen, keine Clientanwendung, ist der Fernzugriff wie im Abschnitt „Fernzugriff auf den Warteschlangenmanager inaktivieren” auf Seite 417 beschrieben zu inaktivieren.

Handelt es sich bei mindestens einer der Anwendungen, die eine Verbindung zum Warteschlangenmanager herstellen sollen, um eine Clientanwendung, muss die ferne Verbindung wie im Abschnitt „Ferne Verbindung zum WS-Manager sichern” auf Seite 409 beschrieben gesichert werden.

In beiden Fällen muss die Verbindungssicherheit wie unter „Verbindungssicherheit einrichten” auf Seite 417 erläutert konfiguriert werden.

Beachten Sie die folgende Tabelle, falls Sie für jeden einzelnen Benutzer, der eine Verbindung zum Warteschlangenmanager herstellt, den Ressourcenzugriff steuern möchten. Ist die Aussage in der ersten Spalte zutreffend, ist die in der zweiten Spalte aufgeführte Maßnahme zu ergreifen.

Anweisung	Maßnahme
Sie verfügen über Anwendungen, die Warteschlangen verwenden	Siehe „Benutzerzugriff auf Warteschlangen steuern” auf Seite 418.
Sie verfügen über Anwendungen, die Themen verwenden	Weitere Informationen finden Sie im Abschnitt „Benutzerzugriff auf Themen steuern” auf Seite 424.
Sie verfügen über Anwendungen, die Abfragen für das WS-Manager-Objekt vornehmen	Weitere Informationen finden Sie im Abschnitt „Berechtigung zum Angeben eines Warteschlangenmanagers erteilen” auf Seite 426.
Sie verfügen über Anwendungen, die Prozessobjekte verwenden	Siehe „Zugriffsberechtigung für Zugriffsprozesse erteilen” auf Seite 427.
Sie verfügen über Anwendungen, die Namenslisten verwenden	Siehe „Berechtigung zum Zugriff auf Namenslisten erteilen” auf Seite 427.

Ferne Verbindung zum WS-Manager sichern

Sie können die ferne Verbindung zum Warteschlangenmanager mit Hilfe von TLS, einem Sicherheitsexit, Kanalauthentifizierungsdatensätzen oder einer Kombination dieser Methoden sichern.

Informationen zu diesem Vorgang

Sie verbinden einen Client mit dem Warteschlangenmanager, indem Sie einen Clientverbindungskanal auf der Client-Workstation und einen Serververbindungskanal auf dem Server verwenden. Sichern Sie solche Verbindungen auf eine der folgenden Arten.

Vorgehensweise

1. TLS mit Kanalauthentifizierungsdatensätzen verwenden:
   a) Verhindern Sie, dass ein definierter Name (DN) einen Kanal öffnet, indem Sie einen SSLPEERMAP-Kanalauthentifizierungssatz verwenden, um alle DNs dem Benutzer USERSRC (NOACCESS) zuzuordnen.
   b) Ermöglichen Sie bestimmten DNs oder DNs, einen Kanal zu öffnen, indem Sie einen SSLPEERMAP-Kanalauthentifizierungsdatensatz verwenden, um sie dem Benutzer USERSRC (CHANNEL) zuzuordnen.
2. TLS mit einem Sicherheitsexit verwenden:
   a) Setzen Sie MCAUSER auf dem Serververbindungskanal auf eine Benutzer-ID ohne Berechtigungen.
   b) Schreiben Sie einen Sicherheitsexit, um einen MCAUSER-Wert zuzuordnen, abhängig von dem Wert des TLS-DN, den er in den Feldern SSLPeerNamePtr und SSLPeerNameLength empfängt, die an den Exit in der MQCD-Struktur übergeben werden.
3. TLS mit festen Kanaldefinitionswerten verwenden:
   a) Legen Sie SSLPEER auf dem Serververbindungskanal auf einen bestimmten Wert oder einen engen Wertebereich fest.
   b) Setzen Sie MCAUSER auf dem Serververbindungskanal auf die Benutzer-ID, mit der der Kanal ausgeführt werden soll.
4. Kanalauthentifizierungsdatensätze für Kanäle verwenden, die TLS nicht verwenden:
   a) Verhindern Sie, dass eine IP-Adresse von den Öffnungskanälen aus verwendet wird. Verwenden Sie dazu einen Kanalauthentifizierungssatz für Adressen-Zuordnungskanal mit ADDRESS (*) und USERSRC (NOACCESS).
   b) Ermöglicht die Verwendung bestimmter IP-Adressen für offene Kanäle unter Verwendung von Adresszuordnungs-Kanalauthentifizierungsdatensätzen für diese Adressen mit USERSRC (CHANNEL).
5. Sicherheitsexit verwenden:
   a) Schreiben Sie einen Sicherheitsexit, um Verbindungen auf der Basis einer beliebigen Eigenschaft zu autorisieren, die Sie auswählen, z. B. die ursprüngliche IP-Adresse.
6. Es ist auch möglich, Kanalauthentifizierungsdatensätze mit einem Sicherheitsexit zu verwenden oder alle drei Methoden zu verwenden, wenn Ihre besonderen Umstände dies erfordern.

Blockieren bestimmter IP-Adressen

Sie können verhindern, dass ein bestimmter Kanal eine eingehende Verbindung von einer IP-Adresse akzeptiert, oder verhindern, dass der gesamte Warteschlangenmanager den Zugriff von einer IP-Adresse aus zulässt, indem ein Kanalauthentifizierungsdatensatz verwendet wird.

Vorbereitende Schritte

Aktivieren Sie die Kanalauthentifizierungsdatensätze, indem Sie den folgenden Befehl ausführen:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Informationen zu diesem Vorgang

Um zu verhindern, dass bestimmte Kanäle eine eingehende Verbindung akzeptieren und sicherstellen, dass Verbindungen nur dann akzeptiert werden, wenn der richtige Kanalname verwendet wird, kann ein Typ von Regel zum Blockieren von IP-Adressen verwendet werden. Wenn Sie eine IP-Adresse für den gesamten Warteschlangenmanager nicht zulassen möchten, verwenden Sie normalerweise eine Firewall, um sie dauerhaft zu blockieren. Es kann jedoch ein anderer Typ von Regel verwendet werden, damit

Sie einige Adressen vorübergehend blockieren können, z. B., wenn Sie darauf warten, dass die Firewall aktualisiert wird.

Prozedur

- Um IP-Adressen für die Verwendung eines bestimmten Kanals zu blockieren, legen Sie einen Kanalauthentifizierungsdatensatz mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record**fest.

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

Der Befehl besteht aus drei Teilen:

SET CHLAUTH (*generic-channel-name*)
Sie verwenden diesen Teil des Befehls, um zu steuern, ob Sie eine Verbindung für den gesamten Warteschlangenmanager, den einzelnen Kanal oder den Bereich der Kanäle blockieren möchten. Was Sie hier einlegen, bestimmt, welche Bereiche abgedeckt werden.

Beispiel:

- SET CHLAUTH('*') -blockiert jeden Kanal in einem Warteschlangenmanager, d.
- SET CHLAUTH ('SYSTEM.*')-blockiert jeden Kanal, der mit SYSTEM beginnt.
- SET CHLAUTH ('SYSTEM.DEF.SVRCONN')-blockiert den Kanal SYSTEM.DEF.SVRCONN

Typ der CHLAUTH-Regel
Verwenden Sie diesen Teil des Befehls, um den Befehlstyp anzugeben, und bestimmt, ob Sie eine einzelne Adresse oder eine Liste von Adressen angeben wollen.

Beispiel:

- TYPE(ADDRESSMAP) -Verwenden Sie ADDRESSMAP, wenn Sie eine einzelne Adresse oder eine Platzhalteradresse angeben möchten. ADDRESS('192.168.*') blockiert z. B. alle Verbindungen, die von einer IP-Adresse stammen, die in 192.168 beginnt.

  Weitere Informationen zum Filtern von IP-Adressen mit Mustern finden Sie unter Generische IP-Adressen .
- TYPE(BLOCKADDR) -Verwenden Sie BLOCKADDR, wenn Sie eine Liste der Adressen angeben wollen, die blockiert werden sollen.

Zusätzliche Parameter
Diese Parameter sind von der Art der Regel abhängig, die Sie im zweiten Teil des Befehls verwendet haben:

- Für TYPE(ADDRESSMAP) verwenden Sie ADDRESS.
- Für TYPE(BLOCKADDR) verwenden Sie ADDRLIST.

Zugehörige Verweise
SET CHLAUTH

Blockierung bestimmter IP-Adressen, wenn der Warteschlangenmanager nicht aktiv ist
Sie können bestimmte IP-Adressen oder Adressbereiche blockieren, wenn der Warteschlangenmanager nicht aktiv ist und Sie daher keine MQSC-Befehle ausgeben können. Sie können IP-Adressen vorübergehend blockieren, indem Sie die `blockaddr.ini`-Datei ändern.

Informationen zu diesem Vorgang

Die Datei `blockaddr.ini` enthält eine Kopie der BLOCKADDR-Definitionen, die vom Queue Manager verwendet werden. Diese Datei wird vom Listener gelesen, wenn der Listener vor dem WS-Manager gestartet wird. Unter diesen Umständen verwendet die Empfangsfunktion alle Werte, die Sie manuell zur Datei `blockaddr.ini` hinzugefügt haben.

Beachten Sie jedoch, dass beim Starten des Queue Manager die Gruppe der BLOCKADDR-Definitionen in die `blockaddr.ini`-Datei geschrieben wird, wobei jede manuelle Bearbeitung überschrieben wird, die Sie möglicherweise ausgeführt haben. Jedes Mal, wenn Sie eine BLOCKADDR-Definition mit dem Befehl **SET CHLAUTH** hinzufügen oder löschen, wird die Datei `blockaddr.ini` aktualisiert. Daher können Sie permanente Änderungen an den BLOCKADDR-Definitionen nur mit dem Befehl **SET CHLAUTH** vornehmen, wenn der Warteschlangenmanager aktiv ist.

Vorgehensweise

1. Öffnen Sie die Datei `blockaddr.ini` in einem Texteditor.

   Die Datei befindet sich im Datenverzeichnis des Warteschlangenmanagers.
2. Fügen Sie IP-Adressen als einfache Schlüsselwort/Wert-Paare hinzu, wobei das Schlüsselwort `Addr` ist.

   Informationen zum Filtern von IP-Adressen mit Mustern finden Sie unter Generische IP-Adressen .

   Beispiel:

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

Zugehörige Tasks

„Blockieren bestimmter IP-Adressen” auf Seite 410
Sie können verhindern, dass ein bestimmter Kanal eine eingehende Verbindung von einer IP-Adresse akzeptiert, oder verhindern, dass der gesamte Warteschlangenmanager den Zugriff von einer IP-Adresse aus zulässt, indem ein Kanalauthentifizierungsdatensatz verwendet wird.

Zugehörige Verweise

SET CHLAUTH

Blockieren bestimmter Benutzer-IDs

Sie können verhindern, dass bestimmte Benutzer einen Kanal verwenden, indem Sie Benutzer-IDs angeben, die, falls sie zugesichert sind, dazu führen, dass der Kanal beendet wird. Geben Sie dazu einen Kanalauthentifizierungsdatensatz an.

Vorbereitende Schritte

Stellen Sie sicher, dass die Kanalauthentifizierungsdatensätze wie folgt aktiviert sind:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Vorgehensweise

Legen Sie mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record**einen Kanalauthentifizierungsdatensatz fest. Sie können z. B. den folgenden MQSC-Befehl ausgeben:

```
SET CHLAUTH(' generic-channel-name ') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.

Die in einem TYPE(BLOCKUSER) bereitgestellte Benutzerliste gilt nur für SVRCONN-Kanäle und nicht für WS-Manager zu WS-Manager-Kanälen.

userID1 und *userID2* sind jeweils die ID eines Benutzers, der verhindert werden soll, dass der Kanal verwendet wird. Sie können auch den Sonderwert `*MQADMIN` angeben, um auf privilegierte Benutzer mit Verwaltungsaufgaben zu verweisen. Weitere Informationen zu privilegierten Benutzern finden Sie

in „Privilegierte Benutzer” auf Seite 357. Weitere Informationen zu *MQADMIN finden Sie unter SET CHLAUTH .

Zugehörige Verweise
SET CHLAUTH

Zuordnung eines fernen Warteschlangenmanagers zu einer MCAUSER-Benutzer-ID
Sie können einen Kanalauthentifizierungsdatensatz verwenden, um das MCAUSER-Attribut eines Kanals entsprechend dem Warteschlangenmanager festzulegen, von dem der Kanal eine Verbindung herstellen soll.

Vorbereitende Schritte

Stellen Sie sicher, dass die Kanalauthentifizierungsdatensätze wie folgt aktiviert sind:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Informationen zu diesem Vorgang

Optional können Sie die IP-Adressen, auf die die Regel angewendet wird, einschränken.

Beachten Sie, dass dieses Verfahren nicht für Serververbindungskanäle gilt. Wenn Sie den Namen eines Serververbindungskanals in den folgenden Befehlen angeben, hat er keine Auswirkungen.

Prozedur

- Legen Sie mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record**einen Kanalauthentifizierungsdatensatz fest. Sie können z. B. den folgenden MQSC-Befehl ausgeben:

  ```
  SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
  ) USERSRC(MAP) MCAUSER(user)
  ```

  generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.
  generic-partner-qmgr-name ist entweder der Name des Warteschlangenmanagers oder ein Muster mit dem Stern (*) als Platzhalterzeichen, das dem Namen des WS-Managers entspricht.
  user ist die Benutzer-ID, die für alle Verbindungen vom angegebenen WS-Manager verwendet werden soll.
- Wenn Sie diesen Befehl auf bestimmte IP-Adressen beschränken möchten, müssen Sie den Parameter **ADDRESS** wie folgt einschließen:

  ```
  SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
  ) USERSRC(MAP) MCAUSER(user) ADDRESS(
  generic-ip-address)
  ```

  generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.
  generic-ip-address ist entweder eine einzelne Adresse oder ein Muster, das den Stern (*) als Platzhalterzeichen oder den Bindestrich (-) enthält, um einen Bereich anzugeben, der mit der Adresse übereinstimmt. Weitere Informationen zu generischen IP-Adressen finden Sie unter Generische IP-Adressen .

Zugehörige Verweise
SET CHLAUTH

Zuordnung einer Clientbenutzer-ID zu einer MCAUSER-Benutzer-ID
Sie können einen Kanalauthentifizierungsdatensatz verwenden, um das MCAUSER-Attribut eines Serververbindungskanals entsprechend der Benutzer-ID zu ändern, die von einem Client empfangen wurde.

Vorbereitende Schritte
Stellen Sie sicher, dass die Kanalauthentifizierungsdatensätze wie folgt aktiviert sind:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Informationen zu diesem Vorgang
Beachten Sie, dass dieses Verfahren nur für Serververbindungskanäle gilt. Es hat keine Auswirkungen auf andere Kanaltypen.

Vorgehensweise

Legen Sie mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record** einen Kanalauthentifizierungsdatensatz fest. Sie können z. B. den folgenden MQSC-Befehl ausgeben:

```
SET CHLAUTH(' generic-channel-name ') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
MCAUSER(
user)
```

generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.

client-user-name ist die Benutzer-ID, die der Clientverbindung zugeordnet ist. Der Wert kann von der Clientanwendung bestätigt werden, die durch die Verbindungsauthentifizierung geändert wird. Verwenden Sie dazu die Option 'early' oder 'set' über einen Kanalexit.

user ist die Benutzer-ID, die anstelle des Clientbenutzernamens verwendet werden soll.

Zugehörige Verweise

SET CHLAUTH

Attribute der Zeilengruppe 'channels' (ChlauthEarlyAdopt)

Zuordnen eines SSL-oder TLS-definierten Namens zu einer MCAUSER-Benutzer-ID
Sie können einen Kanalauthentifizierungsdatensatz verwenden, um das MCAUSER-Attribut eines Kanals entsprechend dem empfangenen definierten Namen (DN) festzulegen.

Vorbereitende Schritte
Stellen Sie sicher, dass die Kanalauthentifizierungsdatensätze wie folgt aktiviert sind:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Vorgehensweise

Legen Sie mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record**einen Kanalauthentifizierungsdatensatz fest. Sie können z. B. den folgenden MQSC-Befehl ausgeben:

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP)
SSLPEER(generic-ssl-peer-name) SSLCERTI(generic-issuer-name)
USERSRC(MAP) MCAUSER(user)
```

generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.

generic-ssl-peer-name ist eine Zeichenfolge, die den IBM MQ-Standardregeln für SSLPEER-Werte folgt. Weitere Informationen finden Sie unter IBM MQ-Regeln für SSLPEER-Werte.

user ist die Benutzer-ID, die für alle Verbindungen mit dem angegebenen DN verwendet werden soll.

generic-issuer-name bezieht sich auf den registrierten Ausstellernamen des Zertifikats, das abgeglichen werden soll. Dieser Parameter ist optional, aber Sie sollten ihn verwenden, um ein falsches Abgleichen des falschen Zertifikats zu vermeiden, wenn mehrere Zertifizierungsstellen im Gebrauch sind.

Zugehörige Verweise

SET CHLAUTH

Zugriff von einem fernen WS-Manager aus sperren

Sie können einen Kanalauthentifizierungsdatensatz verwenden, um zu verhindern, dass ein ferner WS-Manager Kanäle startet.

Vorbereitende Schritte

Stellen Sie sicher, dass die Kanalauthentifizierungsdatensätze wie folgt aktiviert sind:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Informationen zu diesem Vorgang

Beachten Sie, dass dieses Verfahren nicht für Serververbindungskanäle gilt. Wenn Sie den Namen eines Serververbindungskanals im folgenden Befehl angeben, hat er keine Auswirkungen.

Vorgehensweise

Legen Sie mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record**einen Kanalauthentifizierungsdatensatz fest. Sie können z. B. den folgenden MQSC-Befehl ausgeben:

```
SET CHLAUTH(' generic-channel-name ') TYPE(QMGRMAP) QMNAME(' generic-partner-qmgr-name ')
USERSRC(NOACCESS)
```

generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.

generic-partner-qmgr-name ist entweder der Name des Warteschlangenmanagers oder ein Muster mit dem Stern (*) als Platzhalterzeichen, das dem Namen des WS-Managers entspricht.

Zugehörige Verweise

SET CHLAUTH

Blockierung des Zugriffs für eine Clientbenutzer-ID

Sie können einen Kanalauthentifizierungsdatensatz verwenden, um zu verhindern, dass eine Clientbenutzer-ID eine Kanalverbindung aufgebaut hat.

Vorbereitende Schritte

Stellen Sie sicher, dass die Kanalauthentifizierungsdatensätze wie folgt aktiviert sind:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Informationen zu diesem Vorgang

Beachten Sie, dass dieses Verfahren nur für Serververbindungskanäle gilt. Es hat keine Auswirkungen auf andere Kanaltypen.

Vorgehensweise

Legen Sie mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record**einen Kanalauthentifizierungsdatensatz fest. Sie können z. B. den folgenden MQSC-Befehl ausgeben:

```
SET CHLAUTH(' generic-channel-name ') TYPE(USERMAP) CLNTUSER(' client-user-name ') USERSRC(NOAC
CESS)
```

generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.

client-user-name ist die Benutzer-ID, die der Clientverbindung zugeordnet ist. Der Wert kann von der Clientanwendung bestätigt werden, die durch die Verbindungsauthentifizierung geändert wird. Verwenden Sie dazu die Option 'early' oder 'set' über einen Kanalexit.

Zugehörige Verweise

SET CHLAUTH

Blockungszugriff für einen definierten SSL-oder TLS-Namen

Sie können einen Kanalauthentifizierungsdatensatz verwenden, um zu verhindern, dass ein TLS-DN (TLS Distinguished Name, DN) von den Startkanälen entfernt wird.

Vorbereitende Schritte

Stellen Sie sicher, dass die Kanalauthentifizierungsdatensätze wie folgt aktiviert sind:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Vorgehensweise

Legen Sie mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record**einen Kanalauthentifizierungsdatensatz fest. Sie können z. B. den folgenden MQSC-Befehl ausgeben:

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP)
SSLPEER('generic-ssl-peer-name') SSLCERTI(generic-issuer-name)
USERSRC(NOACCESS)
```

generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.

generic-ssl-peer-name ist eine Zeichenfolge, die den IBM MQ-Standardregeln für SSLPEER-Werte folgt. Weitere Informationen finden Sie unter IBM MQ-Regeln für SSLPEER-Werte.

generic-issuer-name bezieht sich auf den registrierten Ausstellernamen des Zertifikats, das abgeglichen werden soll. Dieser Parameter ist optional, aber Sie sollten ihn verwenden, um ein falsches Abgleichen des falschen Zertifikats zu vermeiden, wenn mehrere Zertifizierungsstellen im Gebrauch sind.

Zugehörige Verweise

SET CHLAUTH

Zuordnen einer IP-Adresse zu einer MCAUSER-Benutzer-ID
Sie können einen Kanalauthentifizierungsdatensatz verwenden, um das MCAUSER-Attribut eines Kanals entsprechend der IP-Adresse zu setzen, von der die Verbindung empfangen wird.

Vorbereitende Schritte
Stellen Sie sicher, dass die Kanalauthentifizierungsdatensätze wie folgt aktiviert sind:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Vorgehensweise
Legen Sie mit dem MQSC-Befehl **SET CHLAUTH**oder dem PCF-Befehl **Set Channel Authentication Record**einen Kanalauthentifizierungsdatensatz fest. Sie können z. B. den folgenden MQSC-Befehl ausgeben:

```
SET CHLAUTH(' generic-channel-name ') TYPE(ADDRESSMAP) ADDRESS(' generic-ip-address ')
USERSRC(MAP) MCAUSER(user)
```

generic-channel-name ist entweder der Name eines Kanals, auf den Sie den Zugriff steuern möchten, oder ein Muster, das das Sternsymbol (*) als Platzhalterzeichen enthält, das mit dem Kanalnamen übereinstimmt.
user ist die Benutzer-ID, die für alle Verbindungen mit dem angegebenen DN verwendet werden soll.
generic-ip-address ist entweder die Adresse, von der die Verbindung hergestellt wird, oder ein Muster, das den Stern (*) als Platzhalterzeichen oder den Bindestrich (-) enthält, um einen Bereich anzugeben, der mit der Adresse übereinstimmt.

Zugehörige Verweise
SET CHLAUTH

Fernzugriff auf den Warteschlangenmanager inaktivieren
Inaktivieren Sie den Fernzugriff auf Ihren Warteschlangenmanager, wenn keine Clientanwendungen eine Verbindung zu diesem herstellen sollen.

Informationen zu diesem Vorgang
Die Verbindung von Clientanwendungen zum Warteschlangenmanager kann auf folgende Arten verhindert werden:

Prozedur
- Durch Löschen aller Serververbindungskanäle über den MQSC-Befehl **DELETE CHANNEL**.
- Indem Sie als Nachrichtenkanalagenten-Benutzer-ID (MCAUSER) des Kanals mit dem MQSC-Befehl **ALTER CHANNEL** eine Benutzer-ID ohne Zugriffsrechte definieren.

Verbindungssicherheit einrichten
Erteilen Sie jedem Benutzer oder jeder Gruppe von Benutzern mit einem Geschäftsbedarf die Berechtigung, die Verbindung zum Warteschlangenmanager herzustellen.

Informationen zu diesem Vorgang
Verwenden Sie zum Festlegen der Verbindungssicherheit die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Unter AIX, Linux, and Windows:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
```

- IBM i
  Unter IBM i:

```
GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
```

- z/OS
  Unter z/OS:

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
```

  Mit diesen Befehlen wird Verbindungsberechtigung für Batch-, CICS-, IMS- und Kanalinitiatorverbindungen (CHIN) erteilt. Wenn Sie keinen bestimmten Typ von Verbindung verwenden, lassen Sie die relevanten Befehle weg.

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  Objekt-profil
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  Gruppenname
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Zugehörige Konzepte

„Verbindungssicherheitsprofile für den Kanalinitiator” auf Seite 214
Profile für die Überprüfung von Verbindungen vom Kanalinitiator bestehen aus dem Namen des Warteschlangenmanagers oder der Gruppe mit gemeinsamer Warteschlange, gefolgt von dem Wort *CHIN*. Geben Sie die Benutzer-ID, die vom Kanalinitiator verwendet wird, den Adressraum READ für den Taskadressbereich READ für das Verbindungsprofil an.

Benutzerzugriff auf Warteschlangen steuern

Sie möchten den Anwendungszugriff auf Warteschlangen steuern. In diesem Abschnitt erfahren Sie, wie Sie dazu vorgehen müssen.

Ist die Aussage in der ersten Spalte zutreffend, ist die in der zweiten Spalte aufgeführte Maßnahme zu ergreifen.

Anweisung	Aktion
Die Anwendung ruft Nachrichten aus einer Warteschlange ab	Siehe „Erteilende Berechtigung zum Abrufen von Nachrichten aus Warteschlangen” auf Seite 419.
Die Anwendung definiert Kontext	Siehe „Berechtigung zum Festlegen des Kontexts erteilen” auf Seite 420.

Anweisung	Aktion
Die Anwendung übergibt Kontext	Siehe „Berechtigung zum Übergeben des Kontexts erteilen” auf Seite 421.
Die Anwendung reiht Nachrichten in eine zu einem Cluster gehörigen Warteschlange ein	Siehe „Berechtigung zum Einreihen von Nachrichten in ferne Clusterwarteschlangen berechtigen” auf Seite 511.
Die Anwendung reiht Nachrichten in eine lokale Warteschlange ein	Siehe „Berechtigung zum Eingeben von Nachrichten in eine lokale Warteschlange erteilen” auf Seite 422.
Die Anwendung reiht Nachrichten in eine Modellwarteschlange ein	Siehe „Berechtigung zum Einreihen von Nachrichten in eine Modellwarteschlange erteilen” auf Seite 422.
Die Anwendung reiht Nachrichten in eine ferne Warteschlange ein	Siehe „Berechtigung zum Einlegen von Nachrichten in eine ferne Clusterwarteschlange erteilen” auf Seite 423.

Erteilende Berechtigung zum Abrufen von Nachrichten aus Warteschlangen
Erteilen Sie die Berechtigung zum Abrufen von Nachrichten aus einer Warteschlange oder einer Gruppe von Warteschlangen für jede Gruppe von Benutzern, die einen Geschäftsbedarf für sie haben.

Informationen zu diesem Vorgang

Um die Berechtigung zum Abrufen von Nachrichten aus einigen Warteschlangen zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME(' QMgrName ')
```

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Berechtigung zum Festlegen des Kontexts erteilen
Erteilen Sie dem Benutzer die Berechtigung zum Festlegen des Kontextes für eine Nachricht, die in jede Gruppe von Benutzern mit einem Geschäftsbedarf für diese Nachricht gestellt wird.

Informationen zu diesem Vorgang

Um die Berechtigung zum Festlegen von Kontext in einigen Warteschlangen zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows einen der folgenden Befehle aus:
  - So legen Sie nur den Identitätskontext fest:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
    ```

  - So legen Sie den gesamten Kontext fest:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setall
    ```

  Anmerkung: Um die Berechtigung `setid` oder `setall` verwenden zu können, müssen die Berechtigungen sowohl für das entsprechende Warteschlangenobjekt als auch für das Warteschlangenmanagerobjekt erteilt werden.
- Geben Sie für IBM i einen der folgenden Befehle aus:
  - So legen Sie nur den Identitätskontext fest:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME(' QMgrNa
    me ')
    ```

  - So legen Sie den gesamten Kontext fest:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('
    QMgrName ')
    ```

- Geben Sie für z/OS eine der folgenden Befehlsgruppen aus:
  - So legen Sie nur den Identitätskontext fest:

    ```
    RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
    PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
    ```

  - So legen Sie den gesamten Kontext fest:

    ```
    RDEFINE MQQUEUE QMgrName. ObjectProfile UACC(NONE)
    PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
    ```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Berechtigung zum Übergeben des Kontexts erteilen
Erteilen Sie der Berechtigung, den Kontext aus einer abgerufenen Nachricht an eine Gruppe zu übergeben, die für jede Gruppe von Benutzern mit einem Geschäftsbedarf für sie erforderlich ist.

Informationen zu diesem Vorgang

Um die Berechtigung zum Übergeben von Kontext in einigen Warteschlangen zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- ALW
  Geben Sie für Systeme mit AIX, Linux, and Windows einen der folgenden Befehle aus:
  - Nur Identitätskontext übergeben:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
    ```
  - So übergeben Sie den gesamten Kontext:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
    ```
- IBM i
  Geben Sie für IBM i einen der folgenden Befehle aus:
  - Nur Identitätskontext übergeben:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('
    QMgrName ')
    ```
  - So übergeben Sie den gesamten Kontext:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('
    QMgrName ')
    ```
- z/OS
  Geben Sie für z/OS die folgenden Befehle aus, um den Identitätskontext oder den gesamten Kontext zu übergeben:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Berechtigung zum Eingeben von Nachrichten in eine lokale Warteschlange erteilen
Erteilen Sie der Berechtigung, Nachrichten in eine lokale Warteschlange oder eine lokale Warteschlange zu stellen, jeder Gruppe von Benutzern, die einen Geschäftsbedarf für sie benötigen.

Informationen zu diesem Vorgang

Um die Berechtigung zum Einlegen von Nachrichten in einige lokale Warteschlangen zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Berechtigung zum Einreihen von Nachrichten in eine Modellwarteschlange erteilen
Erteilen Sie der Berechtigung, Nachrichten in eine Modellwarteschlange oder eine Gruppe von Modellwarteschlangen zu stellen, jeder Gruppe von Benutzern, die ein Geschäftsbedarf für sie benötigen.

Informationen zu diesem Vorgang

Modellwarteschlangen werden verwendet, um dynamische Warteschlangen zu erstellen. Sie müssen daher sowohl für das Modell als auch für dynamische Warteschlangen die Berechtigung erteilen. Um diese Berechtigungen zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie auf Systemen mit AIX, Linux, and Windows die folgenden Befehle aus:

```
setmqaut -m QMgrName -n ModelQueueName -t queue -g GroupName +put
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- Geben Sie für IBM i die folgenden Befehle aus:

```
GRTMQMAUT OBJ(' ModelQueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQQUEUE QMgrName.ModelQueueName UACC(NONE)
PERMIT QMgrName.ModelQueueName CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  Modellwarteschlangenname
  Der Name der Modellwarteschlange, auf der dynamische Warteschlangen basieren.

  Objekt-profil
  Der Name der dynamischen Warteschlange oder des generischen Profils, für die Berechtigungen geändert werden sollen.

  Gruppenname
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Berechtigung zum Einlegen von Nachrichten in eine ferne Clusterwarteschlange erteilen
Erteilen Sie der Berechtigung, Nachrichten in eine ferne Clusterwarteschlange oder eine Gruppe von Warteschlangen zu stellen, jeder Gruppe von Benutzern mit einem Geschäftsbedarf dafür.

Informationen zu diesem Vorgang

Wenn Sie eine Nachricht in eine ferne Clusterwarteschlange einlegen möchten, können Sie sie entweder in eine lokale Definition einer fernen Warteschlange oder in eine vollständig qualifizierte ferne Warteschlange stellen. Wenn Sie eine lokale Definition einer fernen Warteschlange verwenden, benötigen Sie die Berechtigung zum Einlegen in das lokale Objekt: siehe „Berechtigung zum Eingeben von Nachrichten in eine lokale Warteschlange erteilen” auf Seite 422. Wenn Sie eine vollständig qualifizierte ferne Warteschlange verwenden, benötigen Sie die Berechtigung, die in die ferne Warteschlange gestellt werden soll. Erteilen Sie diese Berechtigung mit den entsprechenden Befehlen für Ihr Betriebssystem.

Das Standardverfahren besteht darin, eine Zugriffssteuerung für dieSYSTEM.CLUSTER.TRANSMIT.QUEUEdurchzuführen. Beachten Sie, dass dieses Verhalten auch dann gilt, wenn Sie mehrere Übertragungswarteschlangen verwenden.

Das in diesem Abschnitt beschriebene Verfahren gilt nur, wenn Sie das**ClusterQueueAccessControl** Attribut in der qm.iniDatei als *RQMName* konfiguriert haben, wie im Abschnitt Sicherheits-Stanza beschrieben, und den Warteschlangenmanager neu gestartet haben.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -t rqmname -n
ObjectProfile -g GroupName +put
```

  Beachten Sie, dass Sie das Objekt *rqmname* nur für ferne Clusterwarteschlangen verwenden können.
- Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJTYPE(*RMTMQMNAME) OBJ('
ObjectProfile') USER(GroupName) AUT(*PUT) MQMNAME('
QMgrName')
```

  Beachten Sie, dass Sie das RMTMQMNAME-Objekt nur für ferne Clusterwarteschlangen verwenden können.
- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE.ObjectProfile CLASS(MQQUEUE)
ID(GroupName) ACCESS(UPDATE)
```

  Beachten Sie, dass Sie den Namen des fernen Warteschlangenmanagers (oder der Gruppe mit gemeinsamer Warteschlange) nur für ferne Clusterwarteschlangen verwenden können.

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  ObjectProfile
  Der Name des fernen Warteschlangenmanagers oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  GroupName
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Benutzerzugriff auf Themen steuern

Der Zugriff von Anwendungen auf Themen muss kontrolliert werden. In diesem Abschnitt erfahren Sie, wie Sie dazu vorgehen müssen.

Ist die Aussage in der ersten Spalte zutreffend, ist die in der zweiten Spalte aufgeführte Maßnahme zu ergreifen.

Tabelle 71. Benutzerzugriff auf Themen steuern

Anweisung	**Aktion**
Die Anwendung veröffentlicht Nachrichten zu einem Thema	Siehe „Berechtigung zum Publizieren von Nachrichten in einem Thema erteilen” auf Seite 424.
Die Anwendung subskribiert ein Thema	Siehe „Berechtigung zum Subskribieren von Themen erteilen” auf Seite 425.

Berechtigung zum Publizieren von Nachrichten in einem Thema erteilen
Erteilen Sie die Berechtigung zum Publizieren von Nachrichten zu einem Thema oder einer Gruppe von Themen für jede Gruppe von Benutzern, die einen Geschäftsbedarf für sie haben.

Informationen zu diesem Vorgang

Um die Berechtigung zum Publizieren von Nachrichten zu bestimmten Themen zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
```

- Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME(' QMgrNa
me ')
```

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

  Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

  Objekt-profil
  Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

  Gruppenname
  Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Berechtigung zum Subskribieren von Themen erteilen

Erteilen Sie die Berechtigung zum Subskribieren eines Themas oder einer Gruppe von Themen für jede Gruppe von Benutzern, die einen Geschäftsbedarf für sie haben.

Informationen zu diesem Vorgang

Um die Berechtigung zum Subskribieren bestimmter Themen zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME(' QMgrNa
me ')
```

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Berechtigung zum Angeben eines Warteschlangenmanagers erteilen

Erteilen Sie der Berechtigung, einen WS-Manager auf jede Gruppe von Benutzern mit einem Geschäftsbedarf zu stellen.

Informationen zu diesem Vorgang

Um die Berechtigung zum Angeben eines Warteschlangenmanagers zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME(' QMgrName
')
```

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Mit diesen Befehlen wird der Zugriff auf den angegebenen Warteschlangenmanager gewährt. Geben Sie die folgenden Befehle aus, um dem Benutzer die Verwendung des Befehls MQINQ zu ermöglichen:

```
RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Zugriffsberechtigung für Zugriffsprozesse erteilen

Erteilen Sie die Berechtigung für den Zugriff auf einen Prozess oder eine Gruppe von Prozessen für jede Gruppe von Benutzern mit einem Geschäftsbedarf für diese Gruppe.

Informationen zu diesem Vorgang

Um die Berechtigung für den Zugriff auf einige Prozesse zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
```

- Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME(' QMgrName
')
```

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

Berechtigung zum Zugriff auf Namenslisten erteilen

Erteilen Sie die Berechtigung für den Zugriff auf eine Namensliste oder eine Gruppe von Namenslisten für jede Gruppe von Benutzern, die einen Geschäftsbedarf für sie haben.

Informationen zu diesem Vorgang

Um die Berechtigung für den Zugriff auf einige Namenslisten zu erteilen, verwenden Sie die entsprechenden Befehle für Ihr Betriebssystem.

Auf Multiplattformen können Sie auch den Befehl SET AUTHREC verwenden.

Anmerkung: MQ Appliance Unter IBM MQ Appliance können Sie nur den Befehl **SET AUTHREC** verwenden.

Prozedur

- Geben Sie für Systeme mit AIX, Linux, and Windows den folgenden Befehl aus:

```
setmqaut -m QMgrName -n
ObjectProfile -t namelist -g GroupName
+all
```

- Geben Sie für IBM i den folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQNLIST
QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile
CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
```

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Objekt-profil
Der Name des Objekts oder des generischen Profils, für das die Berechtigungen geändert werden sollen.

Gruppenname
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

ALW Berechtigung für die Verwaltung von IBM MQ unter AIX, Linux, and Windows

IBM MQ-Administratoren können alle IBM MQ-Befehle verwenden und Berechtigungen für andere Benutzer erteilen. Wenn Administratoren Befehle an ferne WS-Manager absetzen, müssen sie über die erforderliche Berechtigung auf dem fernen Warteschlangenmanager verfügen. Für Windows-Systeme gelten besondere Einschränkungen.

IBM MQ-Administratoren sind für die Verwendung aller IBM MQ-Befehle berechtigt (einschließlich der Befehle zum Erteilen von IBM MQ-Berechtigungen für andere Benutzer).

Als IBM MQ-Administrator müssen Sie Mitglied der Gruppe **mqm** sein.

Windows Nur unter Windows kann IBM MQ alternativ von lokalen Konten verwaltet werden, wenn diese Mitglieder der Administratorgruppe auf Windows-Systemen sind.

Achtung: Sie können den Benutzer 'Azure AD' mithilfe eines Administratorbefehls der Gruppe 'mqm' hinzufügen. Verwenden Sie zum Beispiel den Befehl `net localgroup mqm AzureAD\<your userID> /add`. Führen Sie anschließend IBM MQ-Verwaltungsbefehle aus oder verwenden Sie IBM MQ Explorer.

Die Gruppe **mqm** wird automatisch erstellt, wenn IBM MQ installiert wird. Sie können der Gruppe weitere Benutzer hinzufügen, damit sie die Verwaltung ausführen können. Alle Mitglieder dieser Gruppe haben

Zugriff auf alle Ressourcen. Dieser Zugriff kann nur widerrufen werden, indem ein Benutzer aus der Gruppe **mqm** entfernt und der Befehl **REFRESH SECURITY** ausgegeben wird.

Administratoren können die Steuerbefehle zum Verwalten von IBM MQ verwenden. Einer dieser Steuerbefehle ist **setmqaut**, mit dem anderen Benutzern die Berechtigungen für den Zugriff auf und die Steuerung von IBM MQ-Ressourcen erteilt werden. Die PCF-Befehle für die Verwaltung von Berechtigungsdatensätzen sind für Nicht-Administratoren verfügbar, denen dsp- und chg-Berechtigungen auf dem Warteschlangenmanager erteilt werden. Weitere Informationen zum Verwalten von Berechtigungen mithilfe von PCF-Befehlen finden Sie im Abschnitt Programmable Command Formats .

Administratoren müssen über die erforderlichen Berechtigungen für die MQSC-Befehle verfügen, die vom fernen WS-Manager verarbeitet werden sollen. Der IBM MQ Explorer gibt PCF-Befehle für die Ausführung von Verwaltungstasks aus. Administratoren benötigen keine weiteren Berechtigungen für die Verwendung des IBM MQ Explorer, um einen Warteschlangenmanager auf dem lokalen System verwalten zu können. Wenn ein Warteschlangenmanager auf einem anderen System vom IBM MQ Explorer verwaltet wird, müssen Administratoren über die erforderlichen Berechtigungen verfügen, damit die PCF-Befehle vom fernen Warteschlangenmanager verarbeitet werden können.

Achtung: Ab IBM MQ 8.0 müssen Sie kein Administrator sein, um den Steuerbefehl **runmqsc** für die Ausgabe von IBM MQ-Scriptbefehlen (MQSC) verwenden zu können.

Wenn **runmqsc** im indirekten Modus verwendet wird, um MQSC-Befehle an einen fernen Warteschlangenmanager zu senden, wird jeder MQSC-Befehl in einen Escape-PCF-Befehl eingebunden.

Weitere Informationen zu Berechtigungsprüfungen bei der Verarbeitung von PCF-und MQSC-Befehlen finden Sie in den folgenden Abschnitten:

- Informationen zu PCF-Befehlen, die für Warteschlangenmanager, Warteschlangen, Prozesse, Namenslisten und Authentifizierungsdatenobjekte ausgeführt werden, finden Sie unter Berechtigung für die Arbeit mit IBM MQ-Objekten. Informationen zu den entsprechenden MQSC-Befehlen, die in Escape-PCF-Befehlen eingebunden sind, finden Sie in diesem Abschnitt.
- Informationen zu PCF-Befehlen, die auf Kanälen, Kanalinitiatoren, Empfangsprogrammen und Clustern ausgeführt werden, finden Sie unter Kanalsicherheit .
- Informationen zu PCF-Befehlen, die für Berechtigungssätze ausgeführt werden, finden Sie unter Berechtigungsprüfung für PCF-Befehle
- z/OS Informationen zu MQSC-Befehlen, die vom Befehlsserver unter IBM MQ for z/OS verarbeitet werden, finden Sie unter Befehlssicherheit und Sicherheit der Befehlsressourcen unter z/OS .

Außerdem hat das Konto SYSTEM auf Windows -Systemen uneingeschränkten Zugriff auf IBM MQ -Ressourcen.

Auf AIX and Linux-Plattformen wird auch die spezielle Benutzer-ID **mqm** erstellt, die nur vom Produkt verwendet wird. Es darf nie für nicht privilegierte Benutzer verfügbar sein. Eigner aller IBM MQ -Objekte ist die Benutzer-ID **mqm**.

Auf Windows -Systemen können Mitglieder der Administratorgruppe ebenso wie das Konto SYSTEM jeden beliebigen Warteschlangenmanager verwalten. Sie können auch eine Domäne **mqm** auf dem Domänencontroller erstellen, die alle privilegierten Benutzer-IDs enthält, die in der Domäne aktiv sind, und fügen Sie sie der lokalen **mqm** -Gruppe hinzu. Einige Befehle wie beispielsweise **crtmqm** bearbeiten Berechtigungen für IBM MQ-Objekte und benötigen daher die Berechtigung für die Verarbeitung dieser Objekte (wie in den folgenden Abschnitten beschrieben). Mitglieder der Gruppe **mqm** haben die Berechtigung zur Arbeit mit allen Objekten, aber auf Windows-Systemen kann unter Umständen der Zugriff verweigert werden, wenn ein lokaler Benutzer und ein in der Domäne authentifizierter Benutzer den gleichen Namen haben. Dieser Vorgang wird im Abschnitt „Principals und Gruppen unter AIX, Linux, and Windows” auf Seite 433 beschrieben.

Windows-Versionen mit der Komponente 'Benutzerkontensteuerung' (User Account Control, UAC) schränkt Aktionen ein, die Benutzer auf bestimmten Funktionen des Betriebssystems ausführen können, selbst dann, wenn es sich dabei um Mitglieder der Administratorgruppe handelt. Wenn Ihre Benutzer-ID in der Administratorgruppe, aber nicht in der Gruppe **mqm** ist, müssen Sie eine Eingabeaufforderung mit erhöhten Rechten zur Ausgabe von IBM MQ-Verwaltungsbefehlen wie **crtmqm** verwenden, da andernfalls

der Fehler `AMQ7077: Sie haben keine Berechtigung zum Ausführen der angeforderten Operation` generiert wird. Um eine erweiterte Eingabeaufforderung zu öffnen, klicken Sie in der Eingabeaufforderung mit der rechten Maustaste auf den Startmenüpunkt oder das Symbol, und wählen Sie **Als Administrator ausführen** aus.

Sie müssen kein Mitglied der **mqm** -Gruppe sein, um die folgenden Aktionen ausführen zu können:

- Geben Sie Befehle von einem Anwendungsprogramm aus, das PCF-Befehle oder MQSC-Befehle in einem Escape-PCF-Befehl absetzt, es sei denn, die Befehle manipulieren Kanalinitiatoren. (Diese Befehle werden in „Kanalinitiatordefinitionen schützen” auf Seite 124 beschrieben.)
- Geben Sie MQI-Aufrufe von einem Anwendungsprogramm aus (es sei denn, Sie möchten die Fast Path-Bindungen im Aufruf `MQCONNX` verwenden) aus.
- Verwenden Sie den Befehl `crtmqcvx` , um ein Codefragment zu erstellen, das die Datenkonvertierung für Datentypstrukturen ausführt.
- Verwenden Sie den Befehl `dspmq` , um die Warteschlangenmanager anzuzeigen.
- Verwenden Sie den Befehl `dspmqtrc`, um eine formatierte IBM MQ-Traceausgabe anzuzeigen.

Eine Einschränkung von 12 Zeichen gilt sowohl für Gruppen-als auch für Benutzer-IDs.

Auf UNIX and Linux-Plattformen ist die Länge von Benutzer-IDs generell auf 12 Zeichen begrenzt. AIX 5.3 hat diesen Grenzwert erhöht, aber IBM MQ hält sich weiterhin an eine 12-Zeichen-Einschränkung auf allen UNIX and Linux-Plattformen. Wenn Sie eine Benutzer-ID mit mehr als 12 Zeichen verwenden, ersetzt IBM MQ diesen Wert durch den Wert `UNKNOWN`. Definieren Sie keine Benutzer-ID mit dem Wert `UNKNOWN`.

ALW Gruppe 'mqm' unter AIX, Linux, and Windows verwalten

Benutzer in der Gruppe 'mqm' verfügen über vollständige Administratorberechtigungen für IBM MQ. Aus diesem Grund sollten Sie keine Anwendungen und normalen Benutzer in der Gruppe 'mqm' registrieren. Die Gruppe 'mqm' sollte nur die Konten der IBM MQ-Administratoren enthalten.

Diese Tasks werden in beschrieben:

- Windows Gruppen unter Windows erstellen und verwalten
- AIX Gruppen unter AIX erstellen und verwalten
- Linux Gruppen unter Linux erstellen und verwalten

Windows Wenn Ihr Domänencontroller unter Windows 2000 oder Windows 2003 oder höher ausgeführt wird, muss Ihr Domänenadministrator möglicherweise ein spezielles Konto für IBM MQ einrichten. Weitere Informationen finden Sie unter IBM MQ mit Prepare IBM MQ Wizard konfigurieren und Windows-Domänenkonten für IBM MQ erstellen und konfigurieren.

ALW Berechtigung zum Arbeiten mit IBM MQ-Objekten in AIX, Linux, and Windows

Alle Objekte werden von IBM MQ geschützt und Prinzipals benötigen für den Zugriff auf diese Objekte die entsprechenden Berechtigungen. Unterschiedliche Principals benötigen unterschiedliche Zugriffsberechtigungen für verschiedene Objekte.

Warteschlangenmanager, Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Authentifizierungsinformationsobjekte werden alle von Anwendungen aufgerufen, die MQI-Aufrufe oder PCF-Befehle verwenden. Diese Ressourcen sind alle durch IBM MQ geschützt und Anwendungen benötigen eine Berechtigung für den Zugriff auf diese Ressourcen. Die Entität, die die Anforderung stellt, kann ein Benutzer, ein Anwendungsprogramm sein, das einen MQI-Aufruf ausgibt, oder ein Verwaltungsprogramm, das einen PCF-Befehl ausgibt. Die Kennung des Anforderers wird als *Principal* bezeichnet.

Verschiedene Gruppen von Principals können verschiedene Typen von Zugriffsberechtigungen für dasselbe Objekt erteilt werden. Für eine bestimmte Warteschlange kann eine Gruppe z. B. sowohl put-als auch get-Operationen ausführen. Eine andere Gruppe ist möglicherweise nur zum Durchsuchen der Warteschlange (`MQGET` mit der Suchoption) berechtigt. In ähnlicher Weise haben einige Gruppen möglicherweise die Berechtigung zum Ändern von Attributen der Warteschlange und zum Ändern der Attribute der Warteschlange oder zum Löschen dieser Warteschlange erhalten.

Einige Operationen sind besonders sensibel und sollten auf privilegierte Benutzer beschränkt sein. Beispiel:

- Zugriff auf einige spezielle Warteschlangen, wie z. B. Übertragungswarteschlangen oder die Befehlswarteschlange SYSTEM.ADMIN.COMMAND.QUEUE
- Programme ausführen, die vollständige MQI-Kontextoptionen verwenden
- Anwendungswarteschlangen erstellen und löschen

Die vollständige Zugriffsberechtigung für ein Objekt wird automatisch der Benutzer-ID, mit der das Objekt erstellt wurde, und allen Mitgliedern der Gruppe 'mqm' sowie den Mitgliedern der lokalen Administratorgruppe auf Windows-Systemen erteilt.

Zugehörige Konzepte

„Berechtigung für die Verwaltung von IBM MQ unter AIX, Linux, and Windows” auf Seite 428
IBM MQ-Administratoren können alle IBM MQ-Befehle verwenden und Berechtigungen für andere Benutzer erteilen. Wenn Administratoren Befehle an ferne WS-Manager absetzen, müssen sie über die erforderliche Berechtigung auf dem fernen Warteschlangenmanager verfügen. Für Windows-Systeme gelten besondere Einschränkungen.

ALW Zeitpunkt für Sicherheitsprüfungen unter AIX, Linux, and Windows

Sicherheitsüberprüfungen werden normalerweise beim Herstellen einer Verbindung zu einem Warteschlangenmanager, beim Öffnen oder Schließen von Objekten und beim Einreihen oder Abrufen von Nachrichten durchgeführt.

Die Sicherheitsprüfungen, die für eine typische Anwendung durchgeführt werden, lauten wie folgt:

Verbindung zum WS-Manager herstellen (MQCONN-oder MQCONNX-Aufrufe)
Dies ist das erste Mal, dass die Anwendung einem bestimmten WS-Manager zugeordnet ist. Der Warteschlangenmanager verknüpft die Betriebsumgebung, um die Benutzer-ID, die der Anwendung zugeordnet ist, zu erkennen. Anschließend prüft IBM MQ, ob die Benutzer-ID berechtigt ist, eine Verbindung zum Warteschlangenmanager herzustellen, und speichert die Benutzer-ID für zukünftige Prüfungen.

Benutzer müssen sich nicht bei IBM MQ anmelden; IBM MQ setzt voraus, dass Benutzer sich beim zugrunde liegenden Betriebssystem angemeldet haben und von diesem authentifiziert wurden.

Das Objekt öffnen (MQOPEN-oder MQPUT1-Aufrufe)
Auf IBM MQ-Objekte wird zugegriffen, indem das Objekt geöffnet wird und Befehle für das Objekt ausgegeben werden. Alle Ressourcenprüfungen werden ausgeführt, wenn das Objekt geöffnet wird, und nicht, wenn tatsächlich auf das Objekt zugegriffen wird. Dies bedeutet, dass die **MQOPEN** -Anforderung den erforderlichen Zugriffstyp angeben muss (z. B. ob der Benutzer nur das Objekt durchsuchen oder eine Aktualisierung durchführen möchte, z. B. Nachrichten in eine Warteschlange einreihen).

IBM MQ überprüft die Ressource, die in der **MQOPEN**-Anforderung angegeben ist. Für einen Aliasnamen oder ein fernes Warteschlangenobjekt ist die verwendete Berechtigung die des Objekts selbst, nicht die Warteschlange, in die der Aliasname oder die ferne Warteschlange aufgelöst wird. Dies bedeutet, dass der Benutzer keine Berechtigung zum Zugriff auf ihn benötigt. Begrenzen Sie die Berechtigung zum Erstellen von Warteschlangen für privilegierte Benutzer. Wenn Sie dies nicht tun, können Benutzer die normale Zugriffssteuerung umgehen, indem Sie einfach einen Aliasnamen erstellen. Wenn eine ferne Warteschlange explizit mit den Namen der Warteschlange und des Warteschlangenmanagers bezeichnet wird, wird die Übertragungswarteschlange, die dem fernen Warteschlangenmanager zugeordnet ist, überprüft.

Die Berechtigung für eine dynamische Warteschlange basiert auf der Basis der Modellwarteschlange, aus der sie abgeleitet wird, ist aber nicht notwendigerweise identisch. Nähere Informationen hierzu finden Sie in Anmerkung „1” auf Seite 144.

Die Benutzer-ID, die vom Warteschlangenmanager für Zugriffsprüfungen verwendet wird, ist die Benutzer-ID, die aus der Betriebsumgebung der Anwendung abgerufen wird, die mit dem Warteschlangenmanager verbunden ist. Eine entsprechend berechtigte Anwendung kann einen **MQOPEN** -Aufruf ausgeben, der eine alternative Benutzer-ID angibt. Anschließend werden Zugriffssteuerungsprüfungen für die alternative Benutzer-ID durchgeführt. Dies ändert nicht die Benutzer-ID, die der Anwendung zugeordnet ist, sondern nur die Benutzer-ID, die für die Prüfungen der Zugriffssteuerung verwendet wird.

Nachrichten einreihen und abrufen (MQPUT-oder MQGET-Aufrufe)
Es werden keine Zugriffssteuerungsprüfungen durchgeführt.

Objekt schließen (MQCLOSE)
Es werden keine Zugriffssteuerungsprüfungen durchgeführt, es sei denn, **MQCLOSE** führt dazu, dass eine dynamische Warteschlange gelöscht wird. In diesem Fall wird geprüft, ob die Benutzer-ID berechtigt ist, die Warteschlange zu löschen.

Subskribieren eines Themas (MQSUB)
Wenn eine Anwendung ein Thema subskribiert, gibt sie die Art der Operation an, die sie ausführen muss. Es wird entweder eine neue Subskription erstellt, eine vorhandene Subskription geändert oder eine vorhandene Subskription wieder aufgenommen, ohne die Subskription zu ändern. Für jeden Typ von Operation prüft der Warteschlangenmanager, ob die Benutzer-ID, die der Anwendung zugeordnet ist, über die Berechtigung zum Ausführen der Operation verfügt.

Wenn eine Anwendung ein Thema subskribiert, werden die Berechtigungsprüfungen für die Themenobjekte durchgeführt, die in der Themenstruktur an oder oberhalb des Punkts in der Themenstruktur gefunden werden, für die die Anwendung subskribiert hat. Die Berechtigungsprüfungen können Prüfungen auf mehr als ein Themenobjekt beinhalten.

Die Benutzer-ID, die der Warteschlangenmanager für die Berechtigungsprüfungen verwendet, ist die Benutzer-ID, die vom Betriebssystem abgerufen wird, wenn die Anwendung eine Verbindung zum WS-Manager herstellt.

Der Warteschlangenmanager führt Berechtigungsprüfungen für Subskribentenwarteschlangen aus, jedoch nicht in den verwalteten Warteschlangen.

ALW Implementierung der Zugriffssteuerung durch IBM MQ unter AIX, Linux, and Windows

IBM MQ verwendet die vom zugrunde liegenden Betriebssystem bereitgestellten Sicherheitsservice mit dem Objektberechtigungsmanager. IBM MQ stellt Befehle bereit, mit denen Zugriffssteuerungslisten erstellt und verwaltet werden.

Eine Schnittstelle für die Zugriffskontrolle mit der Bezeichnung 'Authorization Service Interface' ist Teil von IBM MQ. IBM MQ stellt eine Implementierung eines Zugriffssteuerungsmanagers bereit, der mit der Schnittstelle für den Berechtigungsservice konform ist und als *Objektberechtigungsmanager (OAM)* bezeichnet wird. Dieser Objektberechtigungsmanager wird automatisch für jeden von Ihnen erstellten Warteschlangenmanager installiert und aktiviert, es sei denn, Sie geben eine andere Einstellung vor (siehe „Sicherheitszugriffsprüfungen auf Systemen mit AIX, Linux, and Windows verhindern” auf Seite 388). Der OAM kann von einem beliebigen Benutzer oder einer anderen Anbieterkomponente ersetzt werden, der bzw. die der Berechtigungsserviceschnittstelle entspricht.

Der OAM nutzt die Sicherheitsfunktionen des zugrunde liegenden Betriebssystems unter Verwendung von Betriebssystembenutzer-und Gruppen-IDs aus. Benutzer können nur auf IBM MQ-Objekte zugreifen, wenn sie über die erforderliche Berechtigung verfügen. Im Abschnitt „Zugriff auf Objekte mithilfe des OAM unter AIX, Linux, and Windows steuern” auf Seite 377 wird beschrieben, wie Sie diese Berechtigung erteilen und entziehen.

Der OAM verwaltet eine Zugriffssteuerungsliste (ACL) für jede Ressource, die er steuert. Berechtigungsdaten werden in einer lokalen Warteschlange mit dem Namen SYSTEM.AUTH.DATA.QUEUE gespeichert. Der

Zugriff auf diese Warteschlange ist auf Benutzer in der Gruppe mqm und zusätzlich unter Windowsauf Benutzer in der Gruppe Administratoren und Benutzer, die mit der System-ID angemeldet sind, beschränkt. Der Benutzerzugriff auf die Warteschlange kann nicht geändert werden.

IBM MQ stellt Befehle bereit, mit denen Zugriffssteuerungslisten erstellt und verwaltet werden. Weitere Informationen zu diesen Befehlen finden Sie im Abschnitt „Zugriff auf Objekte mithilfe des OAM unter AIX, Linux, and Windows steuern" auf Seite 377.

IBM MQ übergibt eine Anforderung mit einem Principal, einem Ressourcennamen und einem Zugriffstyp an den OAM. Der OAM erteilt oder verweigert den Zugriff auf der Basis der ACL, die er verwaltet. IBM MQ folgt der Entscheidung des OAM. Wenn der OAM keine Entscheidung treffen kann, verweigert IBM MQ den Zugriff.

ALW Benutzer-ID unter AIX, Linux, and Windows ermitteln

Der Objektberechtigungsmanager gibt den Principal an, der den Zugriff auf eine Ressource anfordert. Die Benutzer-ID, die als Principal verwendet wird, variiert je nach Kontext.

Der Objektberechtigungsmanager (Object Authority Manager, OAM) muss in der Lage sein, zu identifizieren, wer Zugriff auf eine bestimmte Ressource anfordert. In IBM MQ wird der Begriff *Principal* für diese ID verwendet. Der Principal wird eingerichtet, wenn die Anwendung die erste Verbindung zum Warteschlangenmanager herstellt. Sie wird vom Warteschlangenmanager anhand der Benutzer-ID, die der verbundenen Anwendung zugeordnet ist, festgelegt. (Wenn die Anwendung XA-Aufrufe ohne Verbindung zum Warteschlangenmanager absetzt, wird die Benutzer-ID, die der Anwendung zugeordnet ist, die den Aufruf 'xa_open' ausgibt, für Berechtigungsprüfungen durch den Warteschlangenmanager verwendet.)

Auf AIX and Linux-Systemen überprüfen die Berechtigungsprüfroutinen die tatsächlich (angemeldete) Benutzer-ID oder die effektive Benutzer-ID, die der Anwendung zugeordnet ist. Die überprüfte Benutzer-ID kann abhängig vom Bindungstyp sein. Weitere Informationen finden Sie im Abschnitt Installierbare Services .

IBM MQ gibt die Benutzer-ID, die vom System im Nachrichtenheader (MQMD-Struktur) von jeder Nachricht empfangen wird, als die Kennung des Benutzers weiter. Diese Identifikation ist Teil der Nachrichtenkontextinformationen und wird im Abschnitt „Kontextberechtigung unter AIX, Linux, and Windows" auf Seite 436 näher beschrieben. Anwendungen können diese Informationen nur ändern, wenn sie zum Ändern von Kontextinformationen berechtigt sind.

ALW *Principals und Gruppen unter AIX, Linux, and Windows*

Principals können zu Gruppen gehören. Wenn Sie Ressourcenzugriff auf Gruppen und nicht auf Einzelpersonen erteilen, können Sie die erforderliche Verwaltungsmenge reduzieren. Zugriffssteuerungslisten (Access Control Lists, ACLs) basieren auf Gruppen und Benutzer-IDs.

Sie können z. B. eine Gruppe definieren, die aus Benutzern besteht, die eine bestimmte Anwendung ausführen wollen. Anderen Benutzern kann der Zugriff auf alle Ressourcen erteilt werden, die sie benötigen, indem sie ihre Benutzer-ID zur entsprechenden Gruppe hinzufügen.

Dieser Prozess der Definition und Verwaltung von Gruppen wird für bestimmte Plattformen beschrieben:

- AIX Gruppen unter AIX erstellen und verwalten
- Linux Gruppen unter Linux erstellen und verwalten
- Windows Gruppen unter Windows erstellen und verwalten

Ein Principal kann zu mehr als einer Gruppe gehören (sein Gruppensatz). Sie verfügt über die Zusammenfassung aller Berechtigungen, die jeder Gruppe in ihrem Gruppensatz erteilt werden. Diese Berechtigungen werden zwischengespeichert, sodass alle Änderungen, die Sie an der Gruppenzugehörigkeit des Principals vornehmen, erst erkannt werden, wenn der Warteschlangenmanager erneut gestartet wird, es sei denn, Sie geben den MQSC-Befehl **REFRESH SECURITY** (oder dessen PCF-Äquivalent) aus.

Linux AIX Systeme mit AIX and Linux

Ab IBM MQ 8.0 basieren die Zugriffssteuerungslisten (ACLs) auf Benutzer-IDs und Gruppen und können entweder für die Autorisierung verwendet werden, indem das Attribut **SecurityPolicy** auf den entsprechenden Wert gesetzt wird, wie in der Zeilengruppe 'Service' der Datei 'qm.ini' und im Abschnitt Zeilengruppen für Berechtigungsservice unter AIX and Linux konfigurieren beschrieben.

Ab IBM MQ 8.0 können Sie das *benutzerbasierte Modell* zur Berechtigung verwenden, mit dem Benutzer und Gruppen verwendet werden können. Wenn Sie jedoch einen Benutzer im Befehl setmqaut angeben, werden die neuen Berechtigungen nur für diesen Benutzer und nicht für alle Gruppen, zu denen dieser Benutzer gehört, angewendet. Weitere Informationen finden Sie unter OAM-Benutzerberechtigungen auf UNIX- und Linux-Systemen.

Wenn Sie das *gruppenbasierte Modell* für die Berechtigung verwenden, wird die Primärgruppe, zu der die Benutzer-ID gehört, in die Zugriffssteuerungsliste aufgenommen. Die einzelne Benutzer-ID ist nicht enthalten, und die Berechtigung wird allen Mitgliedern dieser Gruppe erteilt. Aus diesem Grund ist zu beachten, dass Sie versehentlich die Berechtigung eines Principals ändern können, indem Sie die Berechtigung eines anderen Principals in derselben Gruppe ändern.

Alle Benutzer sind der Standardbenutzergruppe nobody und standardmäßig keine Berechtigungen für diese Gruppe zugeordnet. Sie können die Berechtigung in der Gruppe nobody ändern, um Benutzern ohne bestimmte Berechtigungen den Zugriff auf IBM MQ-Ressourcen zu erteilen.

V 9.2.1 Ab IBM MQ 9.2.1können Sie die Option `UserExternal` des Attributs **SecurityPolicy** verwenden, um einen Benutzernamen zu erstellen, der kein Betriebssystembenutzername ist. In diesem Fall wird dieser Benutzer mit Ausnahme der Gruppe nobody zu keiner Gruppe gehören. Weitere Informationen zu dieser Option finden Sie unter crtmqm und Zeilengruppe 'Service' der Datei qm.ini ..

Definieren Sie keine Benutzer-ID mit dem Wert `UNKNOWN` . Der Wert `UNKNOWN` wird verwendet, wenn eine Benutzer-ID zu lang ist, so dass beliebige Benutzer-IDs die Zugriffsberechtigungen von `UNKNOWN` verwenden würden.

Unter „Berechtigungen festlegen” auf Seite 442 finden Sie Informationen zur Verwendung von LDAP.

Benutzer-IDs und Gruppennamen können bis zu 12 Zeichen enthalten.

Windows Systeme mit Windows

ACLs basieren sowohl auf Benutzer-IDs als auch auf Gruppen. Die Prüfungen sind unter AIX and Linux identisch. Sie können unterschiedliche Benutzer in verschiedenen Domänen mit derselben Benutzer-ID haben. In IBM MQ können Benutzer-IDs durch einen Domänennamen qualifiziert werden, damit diesen Benutzern verschiedene Zugriffsebenen erteilt werden können.

Der Gruppenname kann optional einen Domänennamen enthalten, der in den folgenden Formaten angegeben wird:

```
GroupName@domain domain_name\group_name
```

Globale Gruppen werden vom OAM nur in zwei Fällen überprüft:

1. Die Zeilengruppe für die WS-Manager-Sicherheit enthält die Einstellung `GroupModel=Global-Groups`. Siehe Securing .
2. Der WS-Manager verwendet eine alternative Sicherheitszugriffsgruppe. Siehe **crtmqm** .

Benutzer-IDs können bis zu 20 Zeichen, Domänennamen bis zu 15 Zeichen und Gruppennamen bis zu 64 Zeichen enthalten.

Der OAM prüft zunächst die lokale Sicherheitsdatenbank, dann die Datenbank der Primärdomäne und schließlich die Datenbank der vertrauenswürdigen Domänen. Die erste Benutzer-ID wird vom OAM für die Überprüfung verwendet. Jede dieser Benutzer-IDs verfügt möglicherweise über unterschiedliche Gruppenzugehörigkeiten auf einem bestimmten Computer.

Mit einigen Steuerbefehlen (z. B. **crtmqm**) werden Berechtigungen in IBM MQ-Objekten mithilfe des Objektberechtigungsmanagers (OAM) geändert. Der OAM durchsucht die Sicherheitsdatenbanken in

der im vorhergehenden Absatz angegebenen Reihenfolge, um die Berechtigungsrechte für eine bestimmte Benutzer-ID zu ermitteln. Daher kann die vom OAM ermittelte Berechtigung die Tatsache außer Kraft setzen, dass eine Benutzer-ID Mitglied der lokalen Gruppe 'mqm' ist. Wenn Sie beispielsweise den Befehl **crtmqm** von einer Benutzer-ID absetzen, die von einem Domänencontroller authentifiziert wird, der über eine globale Gruppe zur lokalen Gruppe 'mqm' gehört, schlägt der Befehl fehl, wenn das System einen lokalen Benutzer mit demselben Namen hat, der nicht zur lokalen Gruppe 'mqm' gehört.

Weitere Informationen zum Festlegen des Attributs **SecurityPolicy** unter Windows finden Sie in den Abschnitten Installierbare Services und Zeilengruppen für Berechtigungsservice unter Windows konfigurieren.

Windows *Windows-Sicherheits-IDs (SIDs)*

IBM MQ unter Windows verwendet die SID, wenn diese verfügbar ist. Wenn mit einer Berechtigungsanforderung keine Windows-SID bereitgestellt wird, identifiziert IBM MQ den Benutzer nur auf Basis des Benutzernamens, was allerdings dazu führen kann, dass die falsche Berechtigung erteilt wird.

Auf Windows-Systems wird die Benutzer-ID durch die Sicherheits-ID (SID) ergänzt. Die SID enthält Informationen, mit denen die vollständigen Benutzerkontodetails in der SAM-Datenbank (Security Account Manager) von Windows angegebenen werden, in der der Benutzer definiert ist. Wenn unter IBM MQ for Windows eine Nachricht erstellt wird, speichert IBM MQ die SID im Nachrichtendeskriptor. Wenn IBM MQ Berechtigungsprüfungen unter Windows ausführt, werden mit der SID die vollständigen Informationen aus der SAM-Datenbank abgefragt. (Die SAM-Datenbank, in der der Benutzer definiert ist, muss zugänglich sein, damit diese Abfrage erfolgreich ausgeführt werden kann.)

Wenn eine Windows-SID nicht mit einer Berechtigungsanforderung bereitgestellt wird, ermittelt IBM MQ den Benutzer standardmäßig nur auf Basis des Benutzernamens. Dies führt dazu, dass die Sicherheitsdatenbanken in der folgenden Reihenfolge durchsucht werden:

1. Die lokale Sicherheitsdatenbank
2. Die Sicherheitsdatenbank der primären Domäne
3. Die Sicherheitsdatenbank der vertrauenswürdigen Domänen

Wenn der Benutzername nicht eindeutig ist, wird möglicherweise eine falsche IBM MQ-Berechtigung erteilt. Um dieses Problem zu vermeiden, schließen Sie in jede Berechtigungsanforderung eine SID ein. Die SID wird von IBM MQ verwendet, um Benutzerberechtigungen zu erstellen.

Um anzugeben, dass alle Berechtigungsanforderungen eine SID enthalten müssen, verwenden Sie **regedit**. Setzen Sie die Sicherheitsrichtlinie auf NTSIDsRequired.

ALW Berechtigung für alternativen Benutzer unter AIX, Linux, and Windows

Sie können angeben, dass eine Benutzer-ID beim Zugriff auf ein IBM MQ-Objekt die Berechtigung eines andern Benutzers verwenden kann. Dies wird als *Berechtigung für alternativen Benutzer* bezeichnet und Sie können sie für jedes IBM MQ-Objekt verwenden.

Die alternative Benutzerberechtigung ist wichtig, wenn ein Server Anforderungen von einem Programm empfängt und sicherstellen will, dass das Programm über die erforderliche Berechtigung für die Anforderung verfügt. Der Server verfügt möglicherweise über die erforderliche Berechtigung, aber er muss wissen, ob das Programm über die Berechtigung für die von ihm angeforderten Aktionen verfügt.

Angenommen, ein Serverprogramm, das unter der Benutzer-ID PAYSERV ausgeführt wird, ruft eine Anforderungsnachricht aus einer Warteschlange ab, die von der Benutzer-ID USER1 in die Warteschlange gestellt wurde. Wenn das Serverprogramm die Anforderungsnachricht abruft, verarbeitet es die Anforderung und versetzt die Antwort zurück in die Warteschlange für Antwortnachrichten, die mit der Anforderungsnachricht angegeben ist. Anstatt die eigene Benutzer-ID (PAYSERV) zu verwenden, um das Öffnen der Warteschlange für Antwortantworten zu autorisieren, kann der Server eine andere Benutzer-ID, in diesem Fall USER1, angeben. In diesem Beispiel können Sie mit der Berechtigung des alternativen Benutzers

steuern, ob PAYSERV als Alternative-Benutzer-ID USER1 angeben darf, wenn die Warteschlange für die Antwortwarteschlange geöffnet wird.

Die alternative Benutzer-ID wird im Feld **AlternateUserId** des Objektdeskriptors angegeben.

Linux Beheben bestimmter Gruppenzugehörigkeitsprobleme in Linux

Einige Systeme geben nur langsam Gruppeninformationen über die normale Reihe von API-Aufrufen des **getgrent** -Betriebssystems zurück. Wenn Ihr Unternehmen Tausende von zu suchenden Gruppen hat und sucht, in welchen Gruppen sich der mqm -Benutzer befindet, kann die langsame Antwort zu einer Zeitlimitüberschreitung des internen Warteschlangenmanagers führen. Zur Umgehung dieses Problems gibt es eine alternative Betriebssystem-API.

Wenn Sie die alternative API, die schneller ist, verwenden möchten, und alle Gruppen aus einem Aufruf zurückgibt, legen Sie die Umgebungsvariable MQS_GETGROUPLIST_API fest.

Möglicherweise wurde ein Fehler von RC2035 empfangen, wenn der Verbindungszugriff auf die sekundäre Gruppe des Benutzers erteilt wurde und die Variable MQS_GETGROUPLIST_API das Problem lindern kann.

IBM MQ verwendet dann die API **getgrouplist** anstelle der API **getgrent** .

So aktivieren Sie **getgrouplist**:

1. Stoppen Sie den Warteschlangenmanager.
2. Setzen Sie den Befehlsexport MQS_GETGROUPLIST_API=1 ab
3. Starten Sie den Warteschlangenmanager erneut.

Wiederholen Sie das Szenario, das fehlgeschlagen ist, und wenn Ihr Problem gelöst wurde, können Sie die Datei `.bashrc / .profile` für den Benutzer `mqm` ändern, um diese Umgebungsvariable hinzuzufügen, oder die Umgebungsvariable zu dem Script hinzufügen, das Sie zum Starten des Warteschlangenmanagers verwenden.

Wenn Ihr System Benutzer- oder Gruppeninformationen für das Betriebssystem aus mehreren Repositorys wie NIS oder LDAP zusammenführt, stellen Sie sicher, dass die Gruppe oder die Benutzer-ID über alle Repositorys konsistent ist, einschließlich der lokalen Repositorys, da diese für die Installation und Festlegung von Berechtigungen auf Betriebssystemebene verwendet werden.

ALW Kontextberechtigung unter AIX, Linux, and Windows

Kontext ist Informationen, die für eine bestimmte Nachricht gelten und in dem Nachrichtendeskriptor (MQMD) enthalten sind, der Teil der Nachricht ist. Anwendungen können die Kontextdaten angeben, wenn entweder ein `MQOPEN` -oder `MQPUT` -Aufruf ausgeführt wird.

Die Kontextinformationen werden in zwei Abschnitten geliefert:

Identitätsabschnitt
: Von wem die Nachricht stammt. Sie setzt sich aus den Feldern `UserIdentifier`, `AccountingToken` und `ApplIdentityData` zusammen.

Ursprungsabschnitt
: Wo die Nachricht herkam und wann sie in die Warteschlange gestellt wurde. Sie setzt sich aus den Feldern `PutApplType`, `PutApplName`, `PutDate`, `PutTime` und `ApplOriginData` zusammen.

Anwendungen können die Kontextdaten angeben, wenn entweder ein `MQOPEN` -oder `MQPUT` -Aufruf ausgeführt wird. Diese Daten können von der Anwendung generiert, von einer anderen Nachricht weitergegeben oder standardmäßig vom Warteschlangenmanager generiert werden. Kontextdaten können beispielsweise von Serverprogrammen verwendet werden, um die Identität des anfordernden Benutzers zu überprüfen und zu testen, ob die Nachricht von einer Anwendung stammt, die unter einer berechtigten Benutzer-ID ausgeführt wird.

Ein Serverprogramm kann die Benutzer-ID von `UserIdentifier` verwenden, um die Benutzer-ID eines alternativen Benutzers zu ermitteln. Sie können die Kontextberechtigung verwenden, um zu steuern,

ob der Benutzer eine beliebige der Kontextoptionen in einem beliebigen Aufruf MQOPEN oder MQPUT1 angeben kann.

In Kontextinformationen steuern finden Sie Informationen zu den Kontextoptionen und eine Übersicht für MQMD zu Beschreibungen der Nachrichtendeskriptorfelder, die sich auf den Kontext beziehen.

Zugriffssteuerung in Sicherheitsexits implementieren

Sie können die Zugriffssteuerung in einem Sicherheitsexit implementieren, indem Sie den MCAUserIdentifier oder den Objektberechtigungsmanager verwenden.

MCAUserIdentifier

Jede Instanz eines Kanals, der aktuell ist, verfügt über eine zugeordnete Kanaldefinitionsstruktur (MQCD). Die Anfangswerte der Felder in MQCD werden durch die Kanaldefinition bestimmt, die von einem IBM MQ-Administrator erstellt wird. Insbesondere wird der Anfangswert eines der Felder *MCAUserIdentifier* bestimmt durch den Wert des Parameters MCAUSER im Befehl DEFINE CHANNEL oder durch das Äquivalent zu MCAUSER, wenn die Kanaldefinition auf andere Weise erstellt wird.

Die MQCD-Struktur wird an ein Kanalexitprogramm übergeben, wenn es von einem MCA aufgerufen wird. Wenn ein Sicherheitsexit von einem MCA aufgerufen wird, kann der Sicherheitsexit den Wert von *MCAUserIdentifier* ändern und einen beliebigen Wert ersetzen, der in der Kanaldefinition angegeben wurde.

Multi Wenn unter Multiplatformsder Wert von *MCAUserIdentifier* nicht leer ist, verwendet der Warteschlangenmanager den Wert von *MCAUserIdentifier* als Benutzer-ID für Berechtigungsprüfungen, wenn ein MCA versucht, auf die Ressourcen des Warteschlangenmanagers zuzugreifen, nachdem er eine Verbindung zum Warteschlangenmanager hergestellt hat. Wenn der Wert von *MCAUserIdentifier* leer ist, verwendet der Warteschlangenmanager stattdessen die Standardbenutzer-ID des MCA. Dies gilt für RCVR-, RQSTR-, CLUSRCVR-und SVRCONN-Kanäle. Zum Senden von Nachrichtenkanalagenten wird die Standardbenutzer-ID immer für Berechtigungsprüfungen verwendet, selbst wenn der Wert von *MCAUserIdentifier* nicht leer ist.

z/OS Unter z/OS kann der Warteschlangenmanager den Wert von *MCAUserIdentifier* für Berechtigungsprüfungen verwenden, sofern er nicht leer ist. Für den Empfang von MCAs und Serververbindungs-MCAs hängt davon ab, ob der Warteschlangenmanager den Wert von *MCAUserIdentifier* für Berechtigungsprüfungen verwendet:

- Der Wert des Parameters PUTAUT in der Kanaldefinition.
- Das für die Prüfungen verwendete RACF-Profil
- Die Zugriffsebene der Benutzer-ID des Kanalinitiatoradressraums in das RESLEVEL-Profil.

Für das Senden von MCAs ist es abhängig von:

- Ob der sendende MCA ein Anrufer oder ein Responder ist
- Die Zugriffsebene der Benutzer-ID des Kanalinitiatoradressraums in das RESLEVEL-Profil.

Die Benutzer-ID, die ein Sicherheitsexit in *MCAUserIdentifier* speichert, kann auf verschiedene Arten erworben werden. Einige Beispiele:

- Ist am Clientende eines MQI-Kanals kein Sicherheitsexit vorhanden, wird eine Benutzer-ID, die der IBM MQ-Clientanwendung zugeordnet ist, vom Nachrichtenkanalagenten der Clientverbindung an den Nachrichtenkanalagenten der Serververbindung gesendet, wenn die Clientanwendung einen MQCONN-Aufruf ausgibt.Die Serververbindung MCA speichert diese Benutzer-ID im Feld *RemoteUserIdentifier* in der Kanaldefinitionsstruktur (MQCD). Wenn der Wert von *MCAUserIdentifier* zu diesem Zeitpunkt leer ist, speichert der MCA die gleiche Benutzer-ID in *MCAUserIdentifier*. Wenn der MCA die Benutzer-ID nicht in *MCAUserIdentifier* speichert, kann ein Sicherheitsexit später ausgeführt werden, indem *MCAUserIdentifier* auf den Wert von *RemoteUserIdentifier* gesetzt wird.

  Tritt die vom Clientsystem gesendete Benutzer-ID in eine andere Sicherheitsdomäne ein, und ist sie auf dem Serversystem ungültig, so kann der Sicherheitsexit diese Benutzer-ID durch eine gültige ersetzen und diese gültige Benutzer-ID im Feld *MCAUserIdentifier* speichern.

- Die Benutzer-ID kann vom Sicherheitsexit der Partnersicherheit in einer Sicherheitsnachricht gesendet werden.

  In einem Nachrichtenkanal kann ein Sicherheitsexit, der von dem sendenden Nachrichtenkanalsystem aufgerufen wird, die Benutzer-ID senden, unter der der sendende Nachrichtenkanalkanalsender ausgeführt wird. Ein Sicherheitsexit, der von dem empfangenden MCA aufgerufen wird, kann dann die Benutzer-ID in *MCAUserIdentifier* speichern. Entsprechend kann ein Sicherheitsexit auf der Clientseite des MQI-Kanals die Benutzer-ID senden, die der IBM MQ MQI client-Anwendung zugeordnet ist. Ein Sicherheitsexit auf dem Serverende des Kanals kann dann die Benutzer-ID in *MCAUserIdentifier* speichern. Wie im vorherigen Beispiel kann der Sicherheitsexit, wenn die Benutzer-ID auf dem Zielsystem nicht gültig ist, die Benutzer-ID für eine gültige Benutzer-ID ersetzen und die ersetzte Benutzer-ID in *MCAUserIdentifier* speichern.

  Wenn ein digitales Zertifikat als Teil des Identifizierungs-und Authentifizierungsservice empfangen wird, kann ein Sicherheitsexit den definierten Namen in dem Zertifikat einer Benutzer-ID zuordnen, die auf dem Zielsystem gültig ist. Anschließend kann die Benutzer-ID in *MCAUserIdentifier* gespeichert werden.
- Wenn TLS auf dem Kanal verwendet wird, wird der definierte Name (DN) des Partners an den Exit im Feld SSLPeerNamePtr von MQCD übergeben, und der DN des Ausstellers dieses Zertifikats wird an den Exit im Feld SSLRemCertIssNamePtr von MQCXP übergeben.

Weitere Informationen über das Feld *MCAUserIdentifier* , die Kanaldefinitionsstruktur, MQCD und die Kanalexitparameterstruktur MQCXP finden Sie unter Channel-Exit-Aufrufe und Datenstrukturen . Weitere Informationen zu der Benutzer-ID, die von einem Clientsystem in einem MQI-Kanal fließt, finden Sie unter Zugriffssteuerung .

Anmerkung: Sicherheitsexitanwendungen, die vor dem Release von IBM WebSphere MQ 7.1 erstellt wurden, müssen möglicherweise aktualisiert werden. Weitere Informationen finden Sie im Abschnitt Kanalsicherheits-Exitprogramme.

Benutzerauthentifizierung für den IBM MQ-Objektberechtigungsmanager

In IBM MQ MQI client-Verbindungen kann mit Sicherheitsexits die MQCSP-Struktur erstellt oder geändert werden, die bei der Benutzerauthentifizierung mit dem Objektberechtigungsmanager (OAM) verwendet wird. Eine Beschreibung hierzu finden Sie im Abschnitt Kanalexitprogramme für Nachrichtenkanäle

Zugriffssteuerung in Nachrichtenexits implementieren

Möglicherweise müssen Sie einen Nachrichtenexit verwenden, um eine Benutzer-ID durch eine andere zu ersetzen.

Betrachten Sie eine Clientanwendung, die eine Nachricht an eine Serveranwendung sendet. Die Serveranwendung kann die Benutzer-ID aus dem Feld *UserIdentifier* im Nachrichtendeskriptor extrahieren und, sofern sie über eine alternative Benutzerberechtigung verfügt, den Warteschlangenmanager anweisen, diese Benutzer-ID für Berechtigungsprüfungen zu verwenden, wenn er für den Client auf IBM MQ-Ressourcen zugreift.

Wenn der Parameter PUTAUT in der Kanaldefinition auf CTX (oder unter z/OS auf ALTMCA) gesetzt ist, wird die Benutzer-ID im Feld *UserIdentifier* der einzelnen eingehenden Nachrichten für Berechtigungsprüfungen verwendet, wenn der Nachrichtenkanalagent die Zielwarteschlange öffnet.

Wenn eine Berichtsnachricht generiert wird, wird unter bestimmten Umständen die Berechtigung der Benutzer-ID in das Feld *UserIdentifier* der Nachricht gesetzt, die den Bericht verursacht. Insbesondere die Berichte zum Bestätigungs-on-Delivery (COD) und das Verfallsdatum werden immer mit dieser Berechtigung versetzt.

Aufgrund dieser Situationen kann es erforderlich sein, eine Benutzer-ID für einen anderen Benutzer im Feld *UserIdentifier* zu ersetzen, wenn eine Nachricht in eine neue Sicherheitsdomäne eintritt. Dies kann durch einen Nachrichtenexit auf der Empfangsseite des Kanals geschehen. Alternativ können Sie sicherstellen, dass die Benutzer-ID im *UserIdentifier* -Feld einer eingehenden Nachricht in der neuen Sicherheitsdomäne definiert ist.

Wenn eine eingehende Nachricht ein digitales Zertifikat für den Benutzer der Anwendung enthält, die die Nachricht gesendet hat, kann ein Nachrichtenexit das Zertifikat überprüfen und den definierten Namen im Zertifikat einer Benutzer-ID zuordnen, die auf dem empfangenden System gültig ist. Anschließend kann das Feld *UserIdentifier* im Nachrichtendeskriptor auf diese Benutzer-ID gesetzt werden.

Wenn es für einen Nachrichtenexit erforderlich ist, um den Wert des Feldes *UserIdentifier* in einer eingehenden Nachricht zu ändern, kann es für den Nachrichtenexit geeignet sein, den Sender der Nachricht gleichzeitig zu authentifizieren. Weitere Informationen finden Sie unter „Identitätsabgleich in Nachrichtenexits” auf Seite 361.

Zugriffssteuerung in API-Exit und API-Steuerübergabeexit implementieren

Ein API-Exit oder ein API-Steuerübergabeexit kann Zugriffssteuerungen bereitstellen, welche die von IBM MQ bereitgestellten ergänzen. Insbesondere kann der Exit die Zugriffssteuerung auf Nachrichtenebene bereitstellen. Der Exit kann sicherstellen, dass eine Anwendung in eine Warteschlange einreiht oder aus einer Warteschlange abgerufen wird, nur die Nachrichten, die bestimmte Kriterien erfüllen.

Betrachten Sie die folgenden Beispiele:

- Eine Nachricht enthält Informationen zu einer Bestellung. Wenn eine Anwendung versucht, eine Nachricht in eine Warteschlange zu stellen, kann ein API-oder API-Steuerübergabeexit prüfen, ob der Gesamtwert der Bestellung kleiner als ein bestimmter Grenzwert ist.
- Nachrichten werden in einer Zielwarteschlange von fernen Warteschlangenmanagern eintreffen. Wenn eine Anwendung versucht, eine Nachricht aus der Warteschlange abzurufen, kann ein API-oder API-Steuerübergabeexit prüfen, ob der Absender der Nachricht berechtigt ist, eine Nachricht an die Warteschlange zu senden.

Multi V 9.2.3 Sicherheit für Streaming-Warteschlangen

Mit der Funktion Streaming-Warteschlangen können Administratoren eine lokale Warteschlange (oder Modellwarteschlange) mit einer sekundären Warteschlange konfigurieren, zu der duplizierte Nachrichten hinzugefügt werden, wenn eine Nachricht zur primären Warteschlange hinzugefügt wird. Bezüglich Berechtigungen zu Streaming-Warteschlangen sind zwei Aspekte zu berücksichtigen.

Berechtigung zur Konfiguration einer Warteschlange zum Streamen von duplizierten Nachrichten

Wenn Sie das Nachrichtenstreaming von duplizierten Nachrichten aus einer Warteschlange in eine sekundäre Warteschlange aktivieren möchten, müssen Sie über die entsprechende Berechtigung verfügen. Die Möglichkeit zum Konfigurieren des Attributs **STREAMQ** für eine Warteschlange erfordert, dass Sie über die folgenden Berechtigungen verfügen:

1. CHG-Berechtigung für die Warteschlange, für die das Attribut **STREAMQ** geändert wird
2. CHG-Berechtigung für die Warteschlange, in die duplizierte Nachrichten eingereiht werden sollen

Die Kombination dieser beiden Berechtigungsprüfungen bei der Konfiguration stellt sicher, dass ein Benutzer, der nur über die CHG-Berechtigung für die ursprüngliche Warteschlange verfügt, keine Nachrichten in eine andere Warteschlange einreihen kann, für die er keine Berechtigungen besitzt.

Berechtigung zum Öffnen der Warteschlange/n und Einreihen von Nachrichten

Wenn eine Anwendung eine Warteschlange öffnet, die mit einer sekundären Warteschlange konfiguriert wurde, wird durch das Attribut **STREAMQ** eine Berechtigungsprüfung durchgeführt, ob der Anwendungsbenutzer über die PUT-Berechtigung für die ursprüngliche Warteschlange verfügt.

Anmerkung: Für den Anwendungsbenutzer in der sekundären Warteschlange wird keine zusätzliche Berechtigungsprüfung durchgeführt. Dies ähnelt dem für Alias-Warteschlangen verwendeten Berechtigungsmodell.

Anwendungen, die Nachrichten entweder nur aus der ursprünglichen oder nur aus der sekundären Warteschlange verarbeiten, erfordern eine GET-oder BROWSE-Berechtigung nur für die Warteschlange, deren Nachrichten sie verarbeiten.

Es werden keine zusätzlichen Berechtigungsprüfungen zum Zeitpunkt der PUT- oder GET-Aktion durchgeführt.

Beispiel

Das folgende Beispiel zeigt die richtigen Berechtigungen, die festgelegt werden müssen, damit der Benutzer `admin` eine ursprüngliche Warteschlange „INQUIRIES.QUEUE“ konfigurieren kann, um ihre duplizierten Nachrichten in die lokale Warteschlange „ANALYTICS.QUEUE“ zu streamen, während der Benutzer `admin` daran gehindert wird, Nachrichten in die Warteschlange „PURCHASES.QUEUE“ zu duplizieren:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(ANALYTICS.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(PURCHASES.QUEUE) PRINCIPAL('admin') AUTHRMV(CHG)
```

Der Benutzer `admin` kann dann den folgenden Befehl ausgeben:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

Aber wenn derselbe Benutzer den folgenden Befehl ausgibt:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

um die Warteschlange „INQUIRIES.QUEUE“ so zu konfigurieren, dass duplizierte Nachrichten in die Warteschlange „PURCHASES.QUEUE“ eingereiht werden, wird der folgende Fehler ausgegeben:

Error TBD

Wenn die Warteschlange „INQUIRIES.QUEUE“ so konfiguriert ist, dass Nachrichten in der Warteschlange „ANALYTICS.QUEUE“ dupliziert werden, werden die folgenden Berechtigungsdatensätze verwendet, um zu erlauben, dass eine Anwendung, die als Benutzer `appuser` ausgeführt wird, Nachrichten in die Warteschlange „INQUIRIES.QUEUE“ sowie duplizierte Nachrichten in die Warteschlange „ANALYTICS.QUEUE“ einreihen kann:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

Anmerkung: Für `appuser` ist kein Berechtigungsdatensatz für „ANALYTICS.QUEUE“. erforderlich. Duplizierte Nachrichten werden vom Warteschlangenmanager in die Warteschlange eingereiht.

Zugehörige Konzepte

Streaming-Warteschlangen

Multi LDAP-Berechtigung

Sie können die LDAP-Berechtigung verwenden, um die Notwendigkeit einer lokalen Benutzer-ID zu entfernen.

Verfügbarkeit der LDAP-Berechtigung auf unterstützten Plattformen

Die LDAP-Berechtigung ist auf Multiplatforms verfügbar:

Achtung:

Ab der allgemeinen Verfügbarkeit von IBM MQ 9.0 ist diese Funktion auf allen Warteschlangenmanagern verfügbar, unabhängig davon, ob sie neu sind oder aus einem früheren Release migriert wurden.

Übersicht über die LDAP-Autorisierung

Mit der LDAP-Berechtigung können Befehle, die die Berechtigungskonfiguration handhaben, wie z. B. **setmqaut** und **DISPLAY AUTHREC**, definierte Namen verarbeiten. Früher wurden Benutzer authentifiziert, indem ihre Berechtigungsnachweise mit den maximal verfügbaren Zeichen verglichen werden, die für Benutzer und Gruppen auf dem lokalen Betriebssystem vorhanden sind.

Achtung: Wenn Sie den Befehl **DEFINE AUTHINFO** ausgeführt haben, müssen Sie den Warteschlangenmanager erneut starten. Wenn Sie den Warteschlangenmanager nicht erneut starten, gibt der Befehl setmqaut nicht das richtige Ergebnis zurück.

Wenn ein Benutzer eine Benutzer-ID und nicht einen definierten Namen (Distinguished Name) bereitstellt, wird die Benutzer-ID verarbeitet. Wenn z. B. eine eingehende Nachricht in einem Kanal mit PUTAUT (CTX) angezeigt wird, werden die Zeichen in der Benutzer-ID einem definierten LDAP-Namen zugeordnet und die entsprechenden Berechtigungsprüfungen werden durchgeführt.

Andere Befehle wie **DISPLAY CONN**funktionieren weiterhin und zeigen den tatsächlichen Wert für die Benutzer-ID an, auch wenn diese Benutzer-ID im lokalen Betriebssystem möglicherweise nicht vorhanden ist.

Linux AIX Wenn die LDAP-Berechtigung vorhanden ist, verwendet der Warteschlangenmanager auf AIX and Linux-Plattformen immer das Benutzermodell der Sicherheit, unabhängig vom Attribut **SecurityPolicy** in der Datei `qm.ini`. Die Festlegung von Berechtigungen für einen einzelnen Benutzer wirkt sich also nur auf diesen Benutzer aus, und nicht auf andere Benutzer, die zu einer dieser Benutzergruppen gehören.

Wie beim Betriebssystemmodell hat ein Benutzer immer noch die kombinierte Berechtigung, die sowohl der Einzelperson als auch allen Gruppen (falls vorhanden) zugeordnet wurde, zu denen der Benutzer gehört.

Nehmen Sie beispielsweise an, dass die folgenden Datensätze in einem LDAP-Repository definiert wurden.

- In der Klasse **inetOrgPerson** :

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- In der Klasse **groupOfNames** :

```
dn="cn=Application Group A, ou=groups, o=yourcompany, c=yourcountry"
        longname=ApplicationGroupA [longer than 12 characters]
        members="cn=JaneDoe, ou=users, o=yourcompany, c=yourcountry",
                "cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
```

Für Authentifizierungszwecke muss ein Warteschlangenmanager, der diesen LDAP-Server verwendet, so definiert worden sein, dass sein **CONNAUTH** -Wert auf ein **AUTHINFO** -Objekt des Typs `IDPWLDAP` verweist und dessen relevante Name-Resolution-Attribute wahrscheinlich wie folgt festgelegt werden:

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

Aufgrund dieser Konfiguration für die Authentifizierung kann eine Anwendung das Feld CSPUserID , das im MQCNO-Aufruf verwendet wird, mit einem der folgenden Werte ausführen:

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

oder

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

In beiden Fällen kann das System die angegebenen Werte verwenden, um den Betriebssystemkontext von " jodoe" zu authentifizieren.

Multi Berechtigungen festlegen

Wie Sie den Kurznamen oder **USRFIELD** verwenden, um Berechtigungen festzulegen.

Die Vorgehensweise beim Arbeiten mit mehreren Formaten, die in „LDAP-Berechtigung” auf Seite 440 beschrieben wird, wird in den Berechtigungsbefehlen mit einer weiteren Erweiterung fortgesetzt, die entweder shortname oder USRFIELD auf einfache Weise verwenden kann.

Die Zeichenfolge gibt ein bestimmtes Attribut im LDAP-Datensatz an, wenn Benutzer (Principals) für die Berechtigung benannt werden.

Wichtig: Die Zeichenfolge darf das Zeichen = nicht enthalten, da dieses Zeichen nicht in einer Betriebssystembenutzer-ID verwendet werden kann.

Wenn Sie einen Principal-Namen an den OAM für die Berechtigung übergeben, die potenziell eine shortname ist, muss die Zeichenfolge in 12 Zeichen passen. Der Zuordnungsalgorithmus versucht zunächst, ihn mit dem Attribut SHORTUSR in seiner LDAP-Abfrage in einen DN aufzulösen.

Wenn dieser Fehler mit einem Fehler UNKNOWN_ENTITY fehlschlägt oder wenn die angegebene Zeichenfolge möglicherweise kein shortname sein kann, wird mit dem Attribut USRFIELD ein weiterer Versuch unternommen, die LDAP-Abfrage zu erstellen.

Achtung: Wenn Sie den Befehl DEFINE AUTHINFO ausgeführt haben, müssen Sie den WS-Manager erneut starten. Wenn Sie den WS-Manager nicht erneut starten, gibt der Befehl setmqaut nicht das richtige Ergebnis zurück.

Für die Verarbeitung von Benutzerberechtigungen sind die folgenden setmqaut -Befehlseinstellungen äquivalent.

Tabelle 72. Benutzerberechtigungseinstellungen

Befehl	**Hinweis**
`setmqaut -m QM -t qmgr -p jodoe +connect`	Dies ist ein flacher, nicht qualifizierter Name, der durch SHORTUSR aufgelöst wird.
`setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect`	Auch ein flacher, nicht qualifizierter Name, der über die USRFIELD-Datei in dieselbe Entität aufgelöst wird.
`setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect`	Verwenden Sie ein benanntes Attribut.
`setmqaut -m QM -t qmgr -p "phone=1234567" +connect`	Es wird ein anderes benanntes Attribut verwendet, das keine der im AUTHINFO-Objekt konfigurierten Attribute sein muss.

Sie können den MQSC-Befehl SET AUTHREC als Alternative zum Befehl **setmqaut** verwenden:

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

oder der PCF-Befehl Set Authority Record (MQCMD_SET_AUTH_REC) mit dem Element MQCACF_PRINCIPAL_ENTITY_NAMES, das die Zeichenfolge enthält:

```
 "cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

Bei der Verarbeitung von Gruppen besteht keine Mehrdeutigkeit bei der `shortname` -Verarbeitung, da es keine Anforderung gibt, in eine beliebige Form eines Gruppennamens in 12 Zeichen einzupassen. Daher gibt es keine Entsprechung des Attributs SHORTUSR für Gruppen.

Dies bedeutet, dass die in Tabelle 73 auf Seite 443 beschriebenen Syntaxbeispiele gültig sind, vorausgesetzt, Sie haben das Objekt AUTHINFO mit den erweiterten Attributen konfiguriert und auf folgende Werte gesetzt:

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

Tabelle 73. Gruppenberechtigungseinstellungen

Befehl	**Hinweis**
`setmqaut -m QM -t qmgr -g ApplicationGroupA +connect`	GRPFIELD zum Auflösen verwenden
`setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect`	Ein einzelnes Attribut benennen
`setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect`	Volldefinierten DN verwenden

Sie können den MQSC-Befehl SET AUTHREC als Alternative zum vorhergehenden Befehl **`setmqaut`** verwenden:

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

oder der PCF-Befehl Set Authority Record (MQCMD_SET_AUTH_REC) mit dem Element MQCACF_GROUP_ENTITY_NAMES, das die Zeichenfolge enthält:

```
 "ApplicationGroupA"
```

Wichtig:

Whichever-Format, das Sie verwenden, um auf einen Namen zu verweisen, unabhängig davon, ob es sich um einen Benutzer oder eine Gruppe handelt, es muss möglich sein, einen eindeutigen DN abzuleiten.

Sie dürfen z. B. nicht über zwei unterschiedliche Datensätze verfügen, die beide den Wert "`shortu=jodoe`" haben.

Wenn ein einzelner eindeutiger DN nicht ermittelt werden kann, gibt der OAM den Wert MQRC_UNKNOWN_ENTITY zurück.

Multi

Autorisierungen anzeigen

Diverse Methoden zum Anzeigen der Berechtigung von Benutzern oder Gruppen.

dspmqaut, Befehl

Die einfachste Methode zum Anzeigen der Berechtigungen, die für einen Benutzer oder eine Gruppe verfügbar sind, besteht darin, den Befehl dspmqaut zu verwenden.

Sie können eine Abfrage in einer der Syntaxvarianten verwenden, um einen Benutzer oder eine Gruppe zu identifizieren. Beachten Sie, dass die Befehlsausgabe die Identität in dem Format wiederholt, das in der Befehlszeile angegeben wurde. Die Ausgabe berichtet nicht über den vollständig aufgelösten DN.

Beispiel:

```
dspmqaut -m QM -t qmgr -p johndoe
Entity johndoe has the following authorizations for object QM:
    connect
```

Oder

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

Befehle 'dmpmqaut' und 'dmpmqcfg'

Der Befehl dmpmqaut und die zugehörigen MQSC- oder PCF-Entsprechungen können den Principal oder die Gruppe in einem der unterstützten Formate angeben, wie in den **setmqaut**-Tabellen unter „Berechtigungen festlegen” auf Seite 442 beschrieben ist. Im Gegensatz zu **dspmqaut** gibt der Befehl **dmpmqaut** jedoch immer den vollständigen DN an.

```
dmpmqaut -m QM -t qmgr -p jodoe
----------------------------------
profile: self
object type:qmgr
entity:cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry
entity type: principal
authority: connect
```

Ebenso zeigt der Befehl dmpmqcfg , der keine Filterung für die ausgewählten Datensätze hat, immer den vollständigen DN in einem Format an, das später erneut wiedergegeben werden kann.

```
dmpmqcfg -m QM -x authrec
----------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

Multi Weitere Überlegungen bei der Verwendung der LDAP-Berechtigung

Eine kurze Beschreibung der Änderungen an der Message Queue Interface (MQI) und anderen MQSC- und PCF-Befehlen, die Sie beachten müssen, wenn Sie die LDAP-Berechtigung von IBM MQ 9.0.0 verwenden.

ADOPTCTX

Es ist nicht erforderlich, dass Anwendungen Authentifizierungsinformationen bereitstellen, oder dass das Attribut ADOPTCTX auf YES gesetzt wird.

Wenn eine Anwendung nicht explizit authentifiziert oder wenn **ADOPTCTX** für das aktive CONNAUTH-Objekt auf NO gesetzt ist, wird der Identitätskontext, der der Anwendung zugeordnet ist, aus der Betriebssystembenutzer-ID übernommen.

Wenn Berechtigungen angewendet werden müssen, wird dieser Kontext einer LDAP-Identität zugeordnet, die dieselben Regeln verwendet wie für die Befehle setmqaut .

Eingabeparameter für MQI-Aufrufe

MQOPEN, MQPUT1 und MQSUB verfügen über Strukturen, mit denen eine alternative Benutzer-ID angegeben werden kann.

Wenn diese Felder verwendet werden, wird die 12 Zeichen lange Benutzer-ID mit denselben Regeln wie in den Befehlen **setmqaut**, **dmpmqaut**und **dspmqaut** einem DN zugeordnet.

MQPUT und MQPUT1 ermöglichen es auch entsprechend berechtigten Programmen, das MQMD-Feld UserIdentifier zu setzen. Der Wert dieses Felds wird während des PUT-Prozesses nicht überwacht und kann auf einen beliebigen Wert gesetzt werden.

Wie üblich kann der Wert **UserIdentifier** jedoch für die Autorisierung in späteren Phasen der Nachrichtenverarbeitung verwendet werden, z. B. wenn PUTAUT (CTX) auf einem Empfangskanal definiert ist.

An diesem Punkt wird die Kennung anhand der Konfiguration des empfangenden WS-Managers, der LDAP oder OS-basiert sein kann, auf die Berechtigung überprüft.

Ausgabeparameter an MQI-Aufrufe

Wenn eine Benutzer-ID einem Programm in einer MQI-Struktur zur Verfügung gestellt wird, handelt es sich um die 12-stellie Kurznamenversion, die der Verbindung zugeordnet ist.

Der **MQAXC.UserId** -Wert für API Exits ist beispielsweise der Kurzname, der aus der LDAP-Zuordnung zurückgegeben wird.

Weitere administrative MQSC-und PCF-Befehle

Befehle, die Benutzerinformationen im Objektstatus anzeigen, wie z. B. DISPLAY CONN USERID, geben den 12-stelligen Kurznamen zurück, der dem Kontext zugeordnet ist. Der vollständige DN wird nicht angezeigt.

Befehle, die die Zusicherung von Identitäten zulassen, wie z. B. die CHLAUTH-Zuordnungsregeln oder MCAUSER -Werte für Kanäle, können Werte bis zu der maximalen Länge annehmen, die für diese Attribute definiert ist (derzeit 64 Zeichen).

Es gibt keine Änderungen an der Syntax. Wenn eine Berechtigung für diese Identität erforderlich ist, wird sie intern unter Verwendung derselben Regeln wie für die Befehle **setmqaut**, **dmpmqaut**und **dspmqaut** einem DN zugeordnet.

Dies bedeutet, dass der MCAUSER-Wert in einer Kanaldefinition möglicherweise nicht als dieselbe Zeichenfolge wie DISPLAY CHSTATUS angezeigt wird, sie sich jedoch auf dieselbe Identität beziehen.

Beispiel:

```
DEFINE CHL(SV1) CHLTYPE(SVRCONN) MCAUSER('cn=JohnDoe')
DEFINE CHL(SV2) CHLTYPE(SVRCONN) MCAUSER('jodoe')
DEFINE CHL(SV3) CHLTYPE(SVRCONN) MCAUSER('JohnDoe1@yourcompany.com')
```

Anschließend DISPLAY CHSTATUS (*) ALL zeigt den Wert für SHORTUSR an, *MCAUSER(jodoe)* für alle Verbindungen.

Multi Zwischen Betriebssystem-und LDAP-Berechtigungsmodellen wechseln

Wie Sie zwischen den verschiedenen Berechtigungsmethoden auf verschiedenen Plattformen wechseln.

Das Attribut CONNAUTH der WS-Manager-Punkte in einem AUTHINFO-Objekt. Wenn das Objekt vom Typ IDPWLDAP ist, wird ein LDAP-Repository für die Authentifizierung verwendet.

Sie können jetzt eine Berechtigungsmethode auf dasselbe Objekt anwenden, die es Ihnen ermöglicht, mit der Betriebssystem-basierten Berechtigung fortzufahren oder mit der LDAP-Berechtigung zu arbeiten.

IBM i, AIX and Linux

IBM i Linux AIX

Der WS-Manager kann jederzeit zwischen OS-und LDAP-Modellen umgeschaltet werden. Sie können die Konfiguration ändern und die aktive Konfiguration mit dem Befehl REFRESH SECURITY TYPE (CONNAUTH) aktivieren.

Wenn dieses Objekt z. B. bereits mit den Verbindungsinformationen für die Authentifizierung konfiguriert wurde:

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

Windows

Windows

Wenn bei einer Änderung der Berechtigungskonfiguration ein Wechsel zwischen Betriebssystem-und LDAP-Modellen erforderlich ist, muss der Warteschlangenmanager erneut gestartet werden, damit die Änderung wirksam wird. Andernfalls können Sie die Änderung aktivieren, indem Sie den Befehl REFRESH SECURITY TYPE (CONNAUTH) verwenden.

Verarbeitungsregeln

Beim Wechsel vom Betriebssystem in die LDAP-Autorisierung werden alle vorhandenen Betriebssystem-berechtigungsregeln, die festgelegt wurden, inaktiv und sind unsichtbar.

Befehle wie **dmpmqaut** zeigen diese Betriebssystemregeln nicht an. In ähnlicher Weise werden bei einer Zurückschaltung von LDAP in das Betriebssystem alle definierten LDAP-Berechtigungen inaktiv und nicht sichtbar, wobei die ursprünglichen Betriebssystemregeln wiederhergestellt werden.

Wenn Sie die Definitionen eines Warteschlangenmanagers aus irgendeinem Grund mit dem Befehl **dmpmqcfg** sichern wollen, enthält diese Sicherung nur die Regeln, die zum Zeitpunkt der Sicherung für die Berechtigungsmethode definiert sind.

Multi LDAP-Verwaltung

Hier finden Sie eine Übersicht über die Verwaltung der einzelnen Plattformen LDAP.

Wenn Sie die LDAP-Berechtigung verwenden, ist die Zugehörigkeit zu der Gruppe `mqm` (oder einem Äquivalent) im Betriebssystem nicht so wichtig. Ein Mitglied dieser Gruppe steuert nur, ob bestimmte Befehlszeilenbefehle verarbeitet werden können.

Sie müssen sich insbesondere in dieser Gruppe befinden, um die Befehle strmqm und endmqm auszugeben.

Sobald der Warteschlangenmanager ausgeführt wird, gibt es jetzt Begrenzungen für das vollständig privilegierte Konto. Abgesehen von der Benutzer-ID der Person, die den Befehl **strmqm** ausgibt, erhalten andere Benutzer, die zur Betriebssystemgruppe `mqm` (oder einer entsprechenden Gruppe) gehören, keine Sonderberechtigungen.

Berechtigungen für andere Benutzer basieren auf den LDAP-Gruppen, zu denen sie gehören. Eine nicht qualifizierte Verwendung des `mqm` -Gruppennamens in Befehlen wie **setmqaut** darf keiner LDAP-Gruppe zugeordnet werden.

AIX and Linux

Linux AIX

Sobald der WS-Manager ausgeführt wird, ist das einzige automatisch voll-privilegierte Konto der Benutzer, der den Warteschlangenmanager gestartet hat.

Die `mqm` -ID ist immer noch vorhanden und wird als Eigner von Betriebssystemressourcen, wie z. B. Dateien, verwendet, da `mqm` die effektive ID ist, unter der der Warteschlangenmanager ausgeführt wird. Der `mqm` -Benutzer kann jedoch nicht automatisch Verwaltungstasks ausführen, die durch den OAM gesteuert werden.

Windows

Windows

Unter Windows handelt es sich bei den automatisch vollständig privilegieren Konten um den Betriebssystembenutzer, der den Warteschlangenmanager gestartet hat, und um den Benutzer, der die zentralen Prozesse des Warteschlangenmanagers wie beispielsweise MUSR_MQADMIN ausführt, wenn der Warteschlangenmanager als Windows-Service gestartet wurde.

Bei der Ausführung im LDAP-Berechtigungsmodus verhält sich Windows auf ähnliche Weise wie AIX and Linux-Plattformen. Es handelt sich um 12 Zeichen kurze Namen und vollständige DNs.

IBM i

IBM i

Unter IBM i werden der Warteschlangenmanager und die QMQM-ID mit den Konten gestartet, die automatisch privilegiert sind.

Sie benötigen beide IDs, da die Benutzer-ID, die den WS-Manager startet, nur zum Starten des Systems erforderlich ist. Sobald die WS-Manager-Prozesse aktiv sind, haben sie nur die Berechtigung QMQM.

Beispielscript für die Bereitstellung von MQADMIN-Berechtigungen

Linux AIX

Da es sinnvoll ist, dass eine Gruppe die vollständige Verwaltung auf einem Warteschlangenmanager vornehmen kann, wird für AIX and Linux-Plattformen ein Beispielscript bereitgestellt:

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

In diesem Beispiel werden zwei Parameter verwendet:

- Name eines Warteschlangenmanagers
- Ein LDAP-Gruppenname

Der Mustercode verarbeitet die Befehle setmqaut und erteilt die vollständige Berechtigung für alle Objekte. Dies ist das gleiche Script, das vom IBM MQ Explorer OAM-Assistenten für Verwaltungsrollen generiert wurde. Der Code beginnt beispielsweise wie folgt:

```
setmqaut -t q -m qmgr -n "**" +alladm -g
      groupname
```

Vertraulichkeit von Nachrichten

Durch das Verschlüsseln von Nachrichten wird sichergestellt, dass die Inhalte von Nachrichten vertraulich bleiben. Es gibt verschiedene Methoden, mit denen Sie Nachrichten in IBM MQ für Ihre Anforderungen verschlüsseln können.

Wenn Sie einen durchgängigen Schutz auf Anwendungsebene für Ihre Point-to-Point-Messaging-Infrastruktur benötigen, können Sie die Nachrichten mit Advanced Message Security verschlüsseln oder Ihren eigenen API-Exit oder einen API-Steuerübergabeexit schreiben.

Die sicherste Lösung ist die Bereitstellung einer End-to-End-Verschlüsselung, bei der eine Nachricht von dem Punkt, an der sie von einer Anwendung eingereiht wird, bis zu dem Punkt, an der sie von der konsumierende Anwendung abgerufen wird, verschlüsselt wird. Dies ist mithilfe von „Advanced Message Security planen” auf Seite 116 (AMS) oder durch das Schreiben eines eigenen API-Exits oder eines API-Steuerübergabeexits möglich; weitere Informationen finden Sie unter „Vertraulichkeit in Benutzerexitprogrammen implementieren” auf Seite 499.

Wenn Sie Nachrichten nur während des Transports durch ein Netz verschlüsseln müssen, können Sie TLS verwenden (siehe „TLS-Sicherheitsprotokolle in IBM MQ” auf Seite 26). Sie können für die Verschlüs-

selung auch Ihren eigenen Sicherheitsexit, Nachrichtenexit oder Sende- und Empfangsexitprogramme schreiben.

V 9.2.0 z/OS Wenn Sie ruhende Nachrichten in einem Warteschlangenmanager verschlüsseln müssen, können Sie die Verschlüsselung von z/OS-Datensätzen für diesen Warteschlangenmanager verwenden. Weitere Informationen finden Sie unter Vertraulichkeit für ruhende Daten unter IBM MQ for z/OS mit Dataset-Verschlüsselung. weitere Informationen hierzu.

Zugehörige Tasks

Verbinden von zwei WS-Managern mit TLS

Client sicher mit einem WS-Manager verbinden

CipherSpecs aktivieren

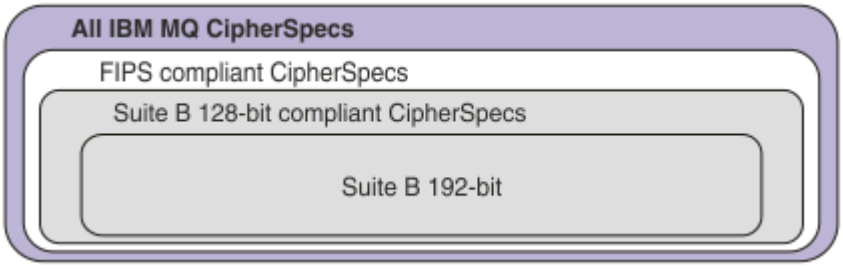
Aktivieren Sie eine CipherSpec mit dem Parameter **SSLCIPH** im MQSC-Befehl **DEFINE CHANNEL** oder **ALTER CHANNEL**.

Anmerkung: Unter AIX, Linux, and Windowsstellt IBM MQ Konformität mit FIPS 140-2 über das Verschlüsselungsmodul "IBM Crypto for C" bereit. Das Zertifikat für dieses Modul wurde in den Langzeitstatus versetzt. Kunden sollten das IBM Crypto for C-Zertifikat anzeigen und sich über Empfehlungen von NIST im Klaren sein. Ein Ersatz-FIPS 140-3-Modul ist derzeit in Bearbeitung und sein Status kann angezeigt werden, indem in der NIST-CMVP-Module in der Prozesslistenach ihm gesucht wird.

Einige der CipherSpecs, die mit IBM MQ verwendet werden können, sind FIPS-konform. Einige der FIPS-konformen CipherSpecs sind auch Suite B-konform, obwohl andere, wie z. B. TLS_RSA_WITH_AES_256_CBC_SHA, nicht vorhanden sind.

Alle mit Suite B kompatiblen CipherSpecs sind ebenfalls FIPS-konform. Alle mit Suite B kompatiblen CipherSpecs fallen in zwei Gruppen: 128 Bit (z. B. ECDHE_ECDSA_AES_128_GCM_SHA256) und 192 Bit (z. B. ECDHE_ECDSA_AES_256_GCM_SHA384).

Das folgende Diagramm veranschaulicht die Beziehung zwischen diesen Untergruppen:



V 9.2.0 V 9.2.0 Ab IBM MQ 9.2.0unterstützt das Produkt das TLS 1.3-Sicherheitsprotokoll auf allen Plattformen. z/OS Unter IBM MQ for z/OS wird TLS 1.3 nur auf z/OS 2.4 oder höher unterstützt.

Die CipherSpecs, die Sie für die jeweiligen Plattformen verwenden können, sind im Abschnitt Tabelle 74 auf Seite 449 aufgeführt. Informationen zur Verwendung dieser CipherSpecs finden Sie unter „TLS 1.3 in IBM MQ verwenden” auf Seite 453 und „IBM MQ MQI client und TLS 1.3” auf Seite 453.

Um die Konfiguration und die zukünftige Migration zu vereinfachen, stellt IBM MQ auch eine Gruppe von Alias-CipherSpecs zur Verfügung. Die Migration vorhandener Sicherheitskonfigurationen für die Verwendung einer Alias-CipherSpec bedeutet, dass Sie Erweiterungen und Unterstützungseinstellungen bei der Verschlüsselung anpassen können, ohne in der Zukunft weitere invasive Konfigurationsänderungen durchführen zu müssen. Diese Alias-CipherSpecs werden im Abschnitt 'Alias CipherSpecs' in Tabelle 74 auf Seite 449 aufgelistet. Weitere Informationen zur Migration zur Verwendung eines Alias-CipherSpec finden Sie im Abschnitt Migrieren vorhandener Sicherheitskonfigurationen für die Verwendung eines Alias-CipherSpec.

V 9.2.0 Sie können die standardmäßigen CipherSpecs wie unter „In IBM MQ aktivierte Standard-CipherSpec-Werte” auf Seite 454 beschrieben konfigurieren. Sie können auch eine alternative Gruppe von CipherSpecs bereitstellen, die für die Verwendung mit Kanälen aktiviert sind:

- Multi IBM MQ for Multiplatforms, wie in „Benutzerdefinierte Liste der sortierten und aktivierten CipherSpecs unter IBM MQ for Multiplatforms bereitstellen” auf Seite 463 beschrieben.
- z/OS IBM MQ for z/OS, wie in „Benutzerdefinierte Liste der sortierten und aktivierten CipherSpecs unter IBM MQ for z/OS bereitstellen” auf Seite 464 beschrieben.

Veraltete CipherSpecs, die Sie bei Bedarf für die Verwendung mit IBM MQ erneut aktivieren können, sind im Abschnitt „Nicht weiter unterstützte CipherSpecs” auf Seite 465 aufgeführt. Informationen zum Aktivieren der veralteten CipherSpecs finden Sie unter „Veraltete CipherSpecs unter IBM MQ for Multiplatforms aktivieren” auf Seite 468 oder „Veraltete CipherSpecs unter z/OS aktivieren” auf Seite 469.

CipherSpecs, die Sie mit TLS-Unterstützung von IBM MQ verwenden können

CipherSpecs, die Sie automatisch mit dem IBM MQ-Warteschlangenmanager verwenden können, werden in der folgenden Tabelle aufgeführt. Wenn Sie ein persönliches Zertifikat anfordern, geben Sie eine Schlüsselgröße für das öffentliche und das private Schlüsselpaar an. Die Schlüsselgröße, die während des TLS-Handshake verwendet wird, ist die Größe, die im Zertifikat gespeichert ist, sofern sie nicht von der CipherSpec bestimmt wird, wie in der Tabelle angegeben.

Tabelle 74. CipherSpecs, die mit TLS-Unterstützung von IBM MQ verwendet werden können

Plattformunterstützung „1” auf Seite 452	**CipherSpec-Name**	**Hexadezimalcode**	**Verwendetes Protokoll**	**MAC-Algorithmus**	**Verschlüsselungsalgorithmus (Verschlüsselungsbits)**	**FIPS** „2” auf Seite 452	**Suite B**
Alias-CipherSpecs							
Alle	ANY_TLS13_OR_HIGHER „3” auf Seite 452 „4” auf Seite 452 „5” auf Seite 452	nicht zutreffend	Vereinbart	Vereinbart	Vereinbart	Vereinbart	Vereinbart
Alle	ANY_TLS13 „4” auf Seite 452 „5” auf Seite 452 „6” auf Seite 452	nicht zutreffend	TLS 1.3	Vereinbart	Vereinbart	Vereinbart	Vereinbart
Alle	ANY_TLS12_OR_HIGHER „4” auf Seite 452 „5” auf Seite 452 „7” auf Seite 452	nicht zutreffend	Vereinbart	Vereinbart	Vereinbart	Vereinbart	Vereinbart
Alle	ANY_TLS12 „8” auf Seite 452	nicht zutreffend	TLS 1.2	Vereinbart	Vereinbart	Vereinbart	Vereinbart
Alle	ANY „9” auf Seite 452	nicht zutreffend	Vereinbart	Vereinbart	Vereinbart	Vereinbart	Vereinbart
CipherSpecs für TLS 1.3							
Alle	TLS_AES_128_GCM_SHA256 „4” auf Seite 452	1301	TLS 1.3	GCM	AES-128 mit GCM (128)	Ja	Nein

Tabelle 74. CipherSpecs, die mit TLS-Unterstützung von IBM MQ verwendet werden können (Forts.)

Plattformunterstützung „1" auf Seite 452	CipherSpec-Name	Hexadezimalcode	Verwendetes Protokoll	MAC-Algorithmus	Verschlüsselungsalgorithmus (Verschlüsselungsbits)	FIPS „2" auf Seite 452	Suite B
Alle	TLS_AES_256_GCM_SHA384 „4" auf Seite 452	1302	TLS 1.3	GCM	AES-256 mit GCM (256)	Ja	Nein
Alle	TLS_CHACHA20_POLY1305_SHA256 „4" auf Seite 452	1303	TLS 1.3	POLY1305	CHA-CHA20 (256)	Nein	Nein
ALW	TLS_AES_128_CCM_SHA256	1304	TLS 1.3	CBC-MAC	AES-128 mit CTR (128)	Ja	Nein
ALW	TLS_AES_128_CCM_8_SHA256 „11" auf Seite 452	1305	TLS 1.3	CBC-MAC	AES-128 mit CTR (128)	Ja	Nein
CipherSpecs für TLS 1.2							
Alle	TLS_RSA_WITH_AES_128_CBC_SHA256 „10" auf Seite 452	003C	TLS 1.2	SHA-256	AES (128)	Ja	Nein
Alle	TLS_RSA_WITH_AES_256_CBC_SHA256 „10" auf Seite 452 „12" auf Seite 452	003D	TLS 1.2	SHA-256	AES (256)	Ja	Nein
Alle	TLS_RSA_WITH_AES_128_GCM_SHA256 „10" auf Seite 452 „13" auf Seite 452	009C	TLS 1.2	SHA-256 und AEAD GCM	AES (128)	Ja	Nein
Alle	TLS_RSA_WITH_AES_256_GCM_SHA384 „10" auf Seite 452 „12" auf Seite 452 „13" auf Seite 452	009D	TLS 1.2	SHA-384 und AEAD GCM	AES (256)	Ja	Nein
Alle	ECDHE_ECDSA_AES_128_CBC_SHA256 „10" auf Seite 452	C023	TLS 1.2	SHA-256	AES (128)	Ja	Nein
Alle	ECDHE_ECDSA_AES_256_CBC_SHA384 „10" auf Seite 452 „12" auf Seite 452	C024	TLS 1.2	SHA-384	AES (256)	Ja	Nein
Alle	ECDHE_RSA_AES_128_CBC_SHA256 „10" auf Seite 452	C027	TLS 1.2	SHA-256	AES (128)	Ja	Nein
Alle	ECDHE_RSA_AES_256_CBC_SHA384 „10" auf Seite 452 „12" auf Seite 452	C028	TLS 1.2	SHA-384	AES (256)	Ja	Nein
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 „12" auf Seite 452 „13" auf Seite 452	C02B	TLS 1.2	SHA-256 und AEAD GCM	AES (SHA384)	Ja	128 Bit
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384 „12" auf Seite 452 „13" auf Seite 452	C02C	TLS 1.2	SHA-384 und AEAD GCM	AES (SHA384)	Ja	192 Bit

Tabelle 74. CipherSpecs, die mit TLS-Unterstützung von IBM MQ verwendet werden können (Forts.)

Plattformunterstützung „1” auf Seite 452	**CipherSpec-Name**	**Hexadezimalcode**	**Verwendetes Protokoll**	**MAC-Algorithmus**	**Verschlüsselungsalgorithmus (Verschlüsselungsbits)**	**FIPS** „2” auf Seite 452	**Suite B**
Alle	ECDHE_RSA_AES_128_GCM_SHA256 „13” auf Seite 452	C02F	TLS 1.2	SHA-256 und AEAD GCM	AES (128)	Ja	Nein
Alle	ECDHE_RSA_AES_256_GCM_SHA384 „12” auf Seite 452 „13” auf Seite 452	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	Ja	Nein

Tabelle 74. CipherSpecs, die mit TLS-Unterstützung von IBM MQ verwendet werden können (Forts.)

Plattformunterstützung „1" auf Seite 452	CipherSpec-Name	Hexadezimalcode	Verwendetes Protokoll	MAC-Algorithmus	Verschlüsselungsalgorithmus (Verschlüsselungsbits)	FIPS „2" auf Seite 452	Suite B

Anmerkungen:

1. Eine Liste der Plattformen, die von den einzelnen Plattformsymbolen abgedeckt werden, finden Sie unter Release- und Plattformsymbole in der Produktdokumentation.
2. Gibt an, ob die CipherSpec auf einer FIPS-zertifizierten Plattform FIPS-zertifiziert ist. Unter Federal Information Processing Standards (FIPS) finden Sie eine Beschreibung des FIPS-Standards.
3. ALW Die Alias-CipherSpec ANY_TLS13_OR_HIGHER vereinbart die höchste Sicherheitsstufe, die das ferne Ende ermöglicht, stellt aber nur über TLS 1.3 oder ein höheres Protokoll eine Verbindung her.
4. z/OS Für die Verwendung von TLS 1.3 oder der CipherSpec ANY in IBM MQ for z/OS muss das Betriebssystem z/OS 2.4 oder höher sein.
5. IBM i Für die Verwendung von TLS 1.3 oder der CipherSpec ANY in IBM i muss die zugrunde liegende Betriebssystemversion TLS 1.3 unterstützen. Weitere Informationen finden Sie unter System TLS support for TLSv1.3.
6. ALW Die Alias-CipherSpec ANY_TLS13 stellt eine Untergruppe zulässiger CipherSpecs dar, die das TLS 1.3-Protokoll verwenden, wie in der folgenden Tabelle für die jeweilige Plattform gezeigt wird.
7. ALW Die Alias-CipherSpec ANY_TLS12_OR_HIGHER vereinbart die höchste Sicherheitsstufe, die das ferne Ende ermöglicht, stellt aber nur über TLS 1.2 oder ein höheres Protokoll eine Verbindung her.
8. Die CipherSpec ANY_TLS12 stellt eine Untergruppe zulässiger CipherSpecs dar, die das TLS 1.2-Protokoll verwenden, wie in der folgenden Tabelle für die jeweilige Plattform gezeigt wird.
9. ALW Die Alias-CipherSpec ANY vereinbart die höchste Sicherheitsstufe, die das ferne Ende ermöglicht.
10. IBM i Diese CipherSpecs sind nicht auf IBM i 7.4-Systemen aktiviert, auf denen der Systemwert QSSLCSLCTL auf *OPSSYS gesetzt ist.
11. ALW Diese CipherSpecs verwenden einen ICV (Integrity Check Value, Wert der Integritätsprüfung) mit 8 Oktett anstelle von 16 Oktett.
12. Eine Verbindung von IBM MQ Explorer zu einem Warteschlangenmanager kann mit dieser CipherSpec nur geschützt werden, wenn die entsprechenden uneingeschränkten Richtliniendateien für die vom Explorer verwendete JRE installiert werden.
13. Windows Linux Gemäß einer Empfehlung von GSKitgilt für TLS 1.2 GCM CipherSpecs die Einschränkung, dass die Verbindung mit der Nachricht AMQ9288Ebeendet wird, nachdem zwei24.5 -TLS-Datensätze unter Verwendung desselben Sitzungsschlüssels gesendet wurden. Diese GCM -Einschränkung ist aktiv, unabhängig vom verwendeten FIPS-Modus.

    Um diesen Fehler zu vermeiden, vermeiden Sie die Verwendung von TLS 1.2 GCM -Verschlüsselungen, aktivieren Sie das Zurücksetzen des geheimen Schlüssels oder starten Sie Ihren IBM MQ -Warteschlangenmanager oder -Client mit der Umgebungsvariablen `GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE` . Für GSKit -Bibliotheken müssen Sie diese Umgebungsvariable auf beiden Seiten der Verbindung festlegen und sie sowohl auf Client-zu-Warteschlangenmanager-Verbindungen als auch auf Warteschlangenmanager-zu-Warteschlangenmanager-Verbindungen anwenden. Beachten Sie, dass sich diese Einstellung auf nicht verwaltete .NET -Clients, jedoch nicht auf Java oder verwaltete .NET Clients auswirkt. Weitere Informationen finden Sie unter AES-GCM -Verschlüsselungseinschränkung.

    Diese Einschränkung gilt nicht für IBM MQ for z/OS.

TLS 1.3 in IBM MQ verwenden

Ab IBM MQ 9.2.0 unterstützt das Produkt TLS 1.3 auf allen Plattformen. Vor IBM MQ 9.2.0 war die TLS 1.3-Unterstützung unter AIX, Linux, and Windows für Continuous Delivery ab IBM MQ 9.1.4 verfügbar.

Warteschlangenmanager, die in IBM MQ 9.2.0 oder höher erstellt sind, unterstützen TLS 1.3 standardmäßig. Für Warteschlangenmanager, die aus früheren Versionen von IBM MQ migriert wurden, muss TLS 1.3 aktiviert sein. Sie können TLS 1.3 in migrierten Warteschlangenmanagern aktivieren, indem Sie die Eigenschaft **AllowTLSV13**=TRUE festlegen:

- Multi Für IBM MQ for Multiplatforms-Warteschlangenmanager bearbeiten Sie die Datei qm.ini und fügen die Eigenschaft **AllowTLSV13**=TRUE unterhalb der SSL-Zeilengruppe hinzu (Verknüpfung zu

```
SSL:
  AllowTLSV13=TRUE
```

- z/OS Bearbeiten Sie für IBM MQ for z/OS-Warteschlangenmanager das QMINI-Data-Set, das in der Start-JCL des Warteschlangenmanagers angegeben ist, und fügen Sie die Eigenschaft **AllowTLSV13**=TRUE unter der Zeilengruppe 'TransportSecurity' hinzu.

```
TransportSecurity:
    AllowTLSV13=TRUE
```

Wenn TLS 1.3 aktiviert ist und bei einer Übereinstimmung mit der TLS 1.3-Spezifikation wird jeder Versuch, mit einer schwachen CipherSpec zu kommunizieren, zurückgewiesen, unabhängig davon, ob sie in IBM MQ aktiviert ist. Die CipherSpecs, die von TLS 1.3 als schwach betrachtet werden, erfüllen eines der folgenden Kriterien:

- Verwendet das SSL 3.0-Protokoll.
- Verwendet RC4 oder RC2 als Verschlüsselungsalgorithmus.
- Hat eine Verschlüsselungsschlüsselgröße (Bit) kleiner-gleich 112.

Diese Einschränkungen sind mit dem Hinweis [3] in Tabelle 1 der veralteten CipherSpecs markiert.

Wenn Sie weiterhin diese CipherSpecs verwenden müssen, müssen Sie den TLS 1.3-Modus inaktivieren:

- ALW Bearbeiten Sie die Datei qm.ini des WS-Managers und ändern Sie die Einstellung der Eigenschaft **AllowTLSV13** wie folgt:

```
SSL:
  AllowTLSV13=FALSE
```

- V 9.2.0 z/OS V 9.2.0 Bearbeiten Sie das QMINI-Dataset des Warteschlangenmanagers und ändern Sie die Einstellung der Eigenschaft **AllowTLSV13** wie folgt:

```
TransportSecurity:
    AllowTLSV13=FALSE
```

IBM MQ MQI client und TLS 1.3

V 9.2.0 ALW

Bei der Verwendung des IBM MQ MQI client wird der Wert von **AllowTLSV13** abgeleitet, es sei denn, er ist in der SSL-Zeilengruppe der Datei mqclient.ini, die von der Anwendung verwendet wird, explizit angegeben.

- Falls schwache CipherSpecs aktiviert sind, wird **AllowTLSV13** auf FALSE gesetzt und es können keine TLS 1.3 CipherSpecs verwendet werden.
- Andernfalls wird **AllowTLSV13** auf TRUE gesetzt und die neuen TLS 1.3 CipherSpecs und Alias-CipherSpecs können verwendet werden.

In IBM MQ aktivierte Standard-CipherSpec-Werte

In der Standardkonfiguration für einen neuen IBM MQ-Warteschlangenmanager stellt IBM MQ die Unterstützung für die TLS 1.2- und TLS 1.3-Protokolle und verschiedene Verschlüsselungsalgorithmen mit CipherSpecs zur Verfügung. Aus Kompatibilitätsgründen kann IBM MQ auch für die Verwendung von SSL 3.0- und TLS 1.0-Protokollen und eine Reihe von Verschlüsselungsalgorithmen konfiguriert werden, die als schwach oder anfällig für Sicherheitslücken bekannt sind. Die Liste der CipherSpecs, die in der Standardkonfiguration aktiviert sind, kann sich ändern, indem die Wartung angewendet wird.

Es ist es möglich, IBM MQ so zu konfigurieren, dass es die Nutzung von CipherSpecs anhand der folgenden Steuerelemente beschränkt oder zulässt:

- Nur FIPS 140-2-konforme CipherSpecs mit SSLFIPS zulassen.
- 
  Nur NSA Suite B-kompatible CipherSpecs mit SUITEB zulassen.
- 
  Angepasste Liste von CipherSpecs mit **AllowedCipherSpecs** zulassen.
- 
  Angepasste Liste von CipherSpecs mit der Umgebungsvariablen **AMQ_ALLOWED_CIPHERS** zulassen.
- 
  Verwendung veralteter CipherSpecs mit der Umgebungsvariable **AllowWeakCipher** oder **AMQ_SSL_WEAK_CIPHER_ENABLE** zulassen.
- 
  Verwenden Sie in der JCL CHINIT die Verwendung von veralteten CipherSpecs, die DD-Anweisungen verwenden.

Anmerkung: Wenn Sie eine angepasste Liste von CipherSpecs mit **AllowedCipherSpecs** oder **AMQ_ALLOWED_CIPHERS** angeben, setzen Sie die Aktivierung aller veralteten CipherSpecs durch. Beachten Sie, dass Sie bei der Verwendung von NSA Suite B-oder FIPS 140-2-Einschränkungen in Kombination mit einer angepassten CipherSpec-Liste sicherstellen müssen, dass die angepasste Liste nur CipherSpecs enthält, die von den Einstellungen für Suite B oder FIPS 140-2 zugelassen sind.

Zugehörige Konzepte

„Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ" auf Seite 49
Dieser Abschnitt enthält Informationen dazu, wie Sie die richtigen CipherSpecs und digitalen Zertifikate für Ihre Sicherheitsrichtlinie auswählen. Dazu wird die Beziehung zwischen CipherSpecs und digitalen Zertifikaten in IBM MQ erläutert.

„CipherSpecs und CipherSuites" auf Seite 21
Kryptografische Sicherheitsprotokolle müssen sich auf die Algorithmen einigen, die von einer sicheren Verbindung verwendet werden. CipherSpecs und CipherSuites definieren bestimmte Kombinationen von Algorithmen.

„IBM MQ für Suite B konfigurieren" auf Seite 47
IBM MQ kann so konfiguriert werden, dass sie in Übereinstimmung mit dem NSA Suite B-Standard auf AIX, Linux, and Windows-Plattformen ausgeführt wird.

„Federal Information Processing Standards (FIPS)" auf Seite 36
In diesem Abschnitt wird das FIPS-Verschlüsselungsprogramm (FIPS Cryptomodule Validation Program) des National Institute of Standards and Technology (US National Institute of Standards and Technology) und die Verschlüsselungsfunktionen eingeführt, die auf TLS-Kanälen verwendet werden können.

Zugehörige Tasks

Vorhandene Sicherheitskonfigurationen für die Verwendung eines Alias-CipherSpec migrieren

Zugehörige Verweise

CHANNEL DEFINE CHANNEL

ALTER CHANNEL

Change, Copy und Create Channel

ALW AES-GCM -Verschlüsselungseinschränkung

Ein Leitfaden zu den Einschränkungen, die für AES-GCM -Verschlüsselungen gelten, wenn sie für TLS-Verschlüsselung verwendet werden Diese Einschränkungen werden von den IETF-und NIST-Organisationen auferlegt und erfordern, dass derselbe Sitzungsschlüssel nicht verwendet werden darf, um mehr als $2^{24.5}$ TLS-Datensätze sicher zu übertragen, wenn AES-GCM -Verschlüsselungen verwendet werden.

Weitere Informationen zu diesen Einschränkungen finden Sie unter RFC 9325 Section 4.4 Limits on Key Usage und RFC 8446 section 5.5.

IBM MQ implementiert die Verschlüsselungsfunktionalität nicht direkt. Stattdessen werden verschiedene Verschlüsselungsbibliotheken verwendet, um TLS-und Advanced Message Security -Funktionalität bereitzustellen. Unter Windows-, Linux-und AIX -Betriebssystemen wird von IBM MQ die Verschlüsselungsbibliothek GSKitverwendet. Bei Anwendungen verwenden die C-Bibliotheken und die nicht verwalteten .NET -Bibliotheken GSKit für die Verschlüsselungsfunktionalität. Die Implementierung der AES-GCM -Verschlüsselungsalgorithmen durch GSKit umfasst die Einschränkungen, die von der Standardgruppe angegeben werden. Außerdem sind diese Einschränkungen standardmäßig aktiviert. Daher wird die IBM MQ TLS-Kommunikation bei Verwendung von AES-GCM -Verschlüsselungen beendet, wenn mehr als $2^{24.5}$ TLS-Datensätze mit demselben Sitzungsschlüssel übertragen werden.

Anmerkung: Diese Einschränkung gilt nicht für IBM i-, IBM Z -oder IBM MQ for HPE NonStop -Plattformen oder Java/JMSverwaltete .NET -Anwendungen, da unterschiedliche Verschlüsselungsbibliotheken verwendet werden und diese Bibliotheken nicht dieselbe Einschränkung implementiert haben.

Wenn ein IBM MQ -Kanal so lange aktiv bleibt, dass mehr als $2^{24.5}$ TLS-Datensätze mit demselben Sitzungsschlüssel übertragen werden, beendet die zugrunde liegende Verschlüsselungsbibliothek die Verbindung. Dadurch wird der Kanal beendet und eine Fehlernachricht AMQ9288E generiert. Anwendungen, deren Kommunikation auf diese Weise beendet wird, empfangen einen `MQRC_CONNECTION_BROKEN` -Rückkehrcode von der IBM MQ -Operation, die ausgeführt wurde.

Die Beendigung der Verbindung kann an beiden Enden der Kommunikation erfolgen, jedoch nur an Enden, die GSKit für die Verschlüsselungsfunktionalität verwenden.

Empfehlung zur Begrenzung der Einschränkung

Es gibt folgende Optionen, wie die aufgrund dieser Einschränkung beendete Kommunikation verhindert oder behandelt werden kann:

Wiederverbindbare Clients verwenden

Anwendungen können so konfiguriert werden, dass sie automatisch versuchen, eine Verbindung herzustellen, wenn eine Verbindung fehlschlägt. Dazu gehören Verbindungen, die aufgrund der Einschränkung GCM beendet werden. Wenn die Clientanwendung für die Verbindungswiederholung konfiguriert ist, wird sie automatisch an jedem Fehlerpunkt zurückgeschrieben und alle Kennungen zum Öffnen von Objekten werden zurückgeschrieben. Dies erfolgt ohne Rückkehr zum Anwendungscode.

Weitere Informationen finden Sie im Abschnitt Automatische Clientverbindungswiederholung.

Rücksetzwert für geheimen Schlüssel festlegen

IBM MQ kann so konfiguriert werden, dass ein Zurücksetzen des Sitzungsschlüssels angefordert wird, nachdem eine konfigurierbare Anzahl von Bytes über einen Kanal übertragen wurde. Wenn dieser Grenzwert erreicht ist, fordert IBM MQ an, dass die Verschlüsselungsschicht einen Sitzungsschlüssel zurücksetzt, was zu einem neuen Sitzungsschlüssel führt.

Es ist wichtig zu beachten, dass der angegebene Wert die Anzahl der übertragenen Byte ist, die sich auf die Größe der von IBM MQgesendeten Nachrichten bezieht. Die Einschränkung gilt für die Anzahl der gesendeten TLS-Datensätze. Es gibt keine direkte Zuordnung zwischen Nachrichtenbytes und TLS-Datensätzen, da ein TLS-Datensatz eine maximale Anzahl von Bytes abhängig von der maximalen Übertragungseinheit (MTU) des Netzes senden kann. Alle gesendeten Nachrichten, die größer als dieser Wert sind, werden als mehrere TLS-Datensätze übertragen. Der MTU-Wert variiert je nach Netz. Es gibt auch andere Gründe, warum ein TLS-Datensatz möglicherweise außerhalb der Übertragung von IBM MQ -Nachrichtendaten gesendet werden muss, z. B. IBM MQ Heartbeatprüfungen, TLS-Alerts oder andere IBM MQ Protokollnachrichten. Diese zusätzlichen TLS-Datensätze zählen zur maximalen

Anzahl von TLS-Datensätzen, werden aber nicht im Wert für das Zurücksetzen des geheimen IBM MQ -Schlüssels gezählt.

Durch regelmäßiges Zurücksetzen eines Sitzungsschlüssels durch Zurücksetzen des geheimen Schlüssels kann verhindert werden, dass der Kanal aufgrund der AES-GCM -Einschränkung beendet wird.

Weitere Informationen finden Sie unter Zurücksetzen von geheimen SSL-und TLS-Schlüsseln.

V 9.2.0 **TLS 1.3 -Verschlüsselungsspezifikationen verwenden**

Während die Einschränkung AES-GCM bei Verwendung des TLS-Protokolls 1.3 weiterhin besteht, unterstützt das TLS-Protokoll 1.3 die automatische Zurücksetzung von Sitzungsschlüsseln, ohne dass die TLS-Kommunikation unterbrochen werden muss. Dadurch kann GSKit das Zurücksetzen des Sitzungsschlüssels verwalten, wenn dies erforderlich ist, ohne dass IBM MQ eine Zurücksetzung des geheimen Schlüssels anfordern muss.

Weitere Informationen finden Sie unter Using TLS 1.3 in IBM MQ in „CipherSpecs aktivieren” auf Seite 448.

Inaktivieren Sie die Einschränkung AES-GCM .

Bei Bedarf kann die Einschränkung durch Festlegen der Umgebungsvariable **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** inaktiviert werden, um die AES-GCM -Einschränkung zu inaktivieren. Dadurch kann eine beliebige Anzahl von TLS-Datensätzen mit demselben Sitzungsschlüssel gesendet werden. Wenn Sie diese Minderung auswählen, muss die Umgebungsvariable an jedem Ende der Kommunikation festgelegt werden, die GSKit für die sichere Kommunikation verwendet.

Warnung: Diese Option wird nicht empfohlen, da Angreifer nach dem Senden von mehr als $2^{24.5}$ TLS-Datensätzen eine Analyse der gesendeten Datensätze durchführen können, um den verwendeten Sitzungsschlüssel zu ermitteln. Sobald der Sitzungsschlüssel ermittelt wurde, wird die gesamte vorhandene und zukünftige Kommunikation, die diesen Sitzungsschlüssel verwendet, beeinträchtigt.

V 9.2.0 V 9.2.0 CipherSpec-Reihenfolge beim TLS-Handshake

Die Reihenfolge von CipherSpecs wird bei der Auswahl aus mehreren möglichen CipherSpecs verwendet, z. B. bei der Verwendung einer der ANY*-CipherSpecs.

Während eines TLS-Handshakes tauschen ein Client und ein Server die unterstützten CipherSpecs und Protokolle in der Reihenfolge der Benutzervorgabe aus. Eine einheitliche CipherSpec, die von beiden Seiten priorisiert wird, wird ausgewählt und für die TLS-Kommunikation verwendet. Bei der Auswahl eines CipherSpec-Protokolls wird auch die Version berücksichtigt. Wenn beispielsweise ein Server TLS 1.2-CipherSpecs vor TLS 1.3-CipherSpecs auflistet, wird trotzdem TLS 1.3 priorisiert, solange der Client diese Version unterstützen kann und über eine einheitliche TLS 1.3-CipherSpec verfügt, die verwendet werden kann.

Ab IBM MQ 9.2.0werden bei der Konfiguration von IBM MQ für TLS die CipherSpecs in der Reihenfolge festgelegt, die in der folgenden Tabelle angezeigt wird (von den am meisten bevorzugten zu den am wenigsten bevorzugten).

Anmerkung: Wenn eine CipherSpec nicht über das Attribut **AllowedCipherSpecs** aktiviert wird, wird es nicht für die Verwendung während eines TLS-Handshakes konfiguriert.

Falls das Attribut **AllowedCipherSpecs** nicht angegeben ist, wird eine Standardliste mit aktivierten Chiffrierwerten verwendet, die in der folgenden Tabelle gezeigt wird.

Tabelle 75. Reihenfolge der CipherSpecs ab IBM MQ 9.2.0

Plattform	CipherSpec	Protokoll	Hexadezimalcode	Standardmäßig aktiviert
Alle	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Ja

Tabelle 75. Reihenfolge der CipherSpecs ab IBM MQ 9.2.0 (Forts.)				
Plattform	**CipherSpec**	**Protokoll**	**Hexadezimalcode**	**Standardmäßig aktiviert**
Alle	TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Ja
Alle	TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Ja
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	1304	Ja
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	1305	Ja
Alle	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Ja
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	C02C	Ja
Alle	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Ja
Alle	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Ja
Alle	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Ja
Alle	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Ja
Alle	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Ja
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	C02B	Ja
Alle	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Ja
Alle	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Ja
Alle	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Ja
Alle	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Ja

Tabelle 75. Reihenfolge der CipherSpecs ab IBM MQ 9.2.0 (Forts.)				
Plattform	**CipherSpec**	**Protokoll**	**Hexadezimalcode**	**Standardmäßig aktiviert**
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	C008	Nein
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	C012	Nein
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	0005	Nein
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	C007	Nein
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	C011	Nein
Alle	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Nein
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	C006	Nein
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	C010	Nein
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	0000	Nein
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Nein
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Nein
IBM i	AES_SHA_US	TLS 1.0	002E	Nein
Alle	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Nein
Alle	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Nein
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	0004	Nein
Alle	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Nein

Tabelle 75. Reihenfolge der CipherSpecs ab IBM MQ 9.2.0 (Forts.)				
Plattform	**CipherSpec**	**Protokoll**	**Hexadezimalcode**	**Standardmäßig aktiviert**
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	0003	Nein
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	0006	Nein
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	0002	Nein
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	0001	Nein
Alle	TRIPLE_DES_SHA_US	SSL v3	000A	Nein
Alle	RC4_SHA_US	SSL v3	0005	Nein
Alle	RC4_MD5_US	SSL v3	0004	Nein
Alle	DES_SHA_EXPORT	SSL v3	0009	Nein
Alle	RC4_MD5_EXPORT	SSL v3	0003	Nein
Alle	RC2_MD5_EXPORT	SSL v3	0006	Nein
Alle	NULL_SHA	SSL v3	0002	Nein
Alle	NULL_MD5	SSL v3	0001	Nein
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	FEFF	Nein
ALW	RC4_56_SHA_EXPORT1024	SSL v3	0064	Nein
ALW	DES_SHA_EXPORT1024	SSL v3	0062	Nein
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	FEFE	Nein

Diese Liste wurde erstellt, indem die Protokolle mit der Standardliste sortiert werden, die von der von IBM MQ auf z/OS verwendeten Verschlüsselungsbibliothek bereitgestellt wird und über z/OS und verteilte Plattformen hinweg konsistent ist.

Reihenfolge ändern

Wenn eine andere Reihenfolge gewünscht wird, kann eine neue Reihenfolge von CipherSpecs mit dem Attribut **AllowedCipherSpecs** der SSL-Zeilengruppe unter IBM MQ for Multiplatforms z/OS oder der TransportSecurity-Zeilengruppe unter IBM MQ for z/OS bereitgestellt werden, wobei die folgenden Regeln gelten:

- Es werden immer höhere Protokollversionen verwendet, unabhängig von ihrer Position in der Liste.
- Inaktivierte CipherSpecs werden erneut aktiviert, wenn sie in der Liste angegeben werden.
- Die Listenreihenfolge des TLS-Servers hat eine höhere Priorität als der TLS-Client.
- Wenn TLS 1.3 aktiviert ist, werden bestimmte CipherSpecs nicht unterstützt.

Beispielsweise ist in IBM MQ for Multiplatforms im Warteschlangenmanager Folgendes konfiguriert:

```
SSL:
AllowedCipher
Specs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_RSA_WITH
_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

z/OS In IBM MQ for z/OS ist im Warteschlangenmanager Folgendes konfiguriert:

```
TransportSecurity:
AllowedCipher
Specs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_RSA_WITH
_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

In diesem Fall gilt Folgendes:

- Ein Client, der eine Verbindung mit ANY_TLS12 herstellt, verwendet wahrscheinlich die TLS 1.2-CipherSpec TLS_RSA_WITH_AES_128_GCM_SHA256.
- Ein Client, der eine Verbindung mit ANY_TLS12_OR_HIGHER herstellt, verwendet wahrscheinlich die TLS 1.3-CipherSpec TLS_AES_128_GCM_SHA256 (vorausgesetzt, der Client unterstützt TLS 1.3).
- Ein Client, der eine Verbindung mit der TLS 1.0-CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA herstellt, verwendet diese CipherSpec.

Vorgängerversionen von IBM MQ

Vor IBM MQ 9.2.0 wurde die folgende Reihenfolge für CipherSpecs verwendet:

Tabelle 76. Reihenfolge der CipherSpecs vor IBM MQ 9.2.0

Plattform	CipherSpec	Protokoll	Standardmäßig aktiviert
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	Nein
IBM i	AES_SHA_US	TLS 1.0	Nein
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	Nein
Alle	RC4_SHA_US	SSL v3	Nein
Alle	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	Nein
Alle	RC4_MD5_US	SSL v3	Nein
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	Nein
Alle	TRIPLE_DES_SHA_US	SSL v3	Nein
Alle	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	Nein
ALW	DES_SHA_EXPORT1024	SSL v3	Nein
Alle	RC4_56_SHA_EXPORT1024	SSL v3	Nein

Tabelle 76. Reihenfolge der CipherSpecs vor IBM MQ 9.2.0 (Forts.)			
Plattform	**CipherSpec**	**Protokoll**	**Standardmäßig aktiviert**
Alle	RC4_MD5_EXPORT	SSL v3	Nein
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	Nein
Alle	RC2_MD5_EXPORT	SSL v3	Nein
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	Nein
Alle	DES_SHA_EXPORT	SSL v3	Nein
Alle	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	Nein
Alle	NULL_SHA	SSL v3	Nein
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	Nein
Alle	NULL_MD5	SSL v3	Nein
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	Nein
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	Nein
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	Nein
Alle	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	Ja
Alle	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	Ja
Alle	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	Nein
Alle	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	Ja
Alle	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	Ja
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	Nein
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	Nein
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	Nein

Tabelle 76. Reihenfolge der CipherSpecs vor IBM MQ 9.2.0 (Forts.)			
Plattform	**CipherSpec**	**Protokoll**	**Standardmäßig aktiviert**
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	Nein
Alle	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	Ja
Alle	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	Ja
Alle	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	Ja
Alle	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	Ja
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	Ja
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	Ja
Alle	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	Ja
Alle	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	Ja
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	Nein
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	Nein
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	Nein
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	Nein
Multi	TLS_AES_128_GCM_SHA256	TLS 1.3	Ja
Multi	TLS_AES_256_GCM_SHA384	TLS 1.3	Ja
Multi	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	Ja
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	Ja

Tabelle 76. Reihenfolge der CipherSpecs vor IBM MQ 9.2.0 (Forts.)			
Plattform	**CipherSpec**	**Protokoll**	**Standardmäßig aktiviert**
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	Ja

Wichtig: Ab dem 23. Juli 2020 werden mit dem folgenden AllowedCipherSpecs-Attribut nur CipherSpecs aktiviert, die derzeit standardmäßig aktiviert sind. Sie sollten allerdings die aktivierten CipherSpecs mit dem folgenden AllowedCipherSpecs-Attribut mit aktuellen Daten prüfen, um sicherzustellen, dass CipherSpecs, die seit diesem Datum nicht mehr unterstützt werden, nicht versehentlich erneut aktiviert werden.

Wenn Sie zu dieser Reihenfolge der CipherSpecs zurückkehren müssen, können Sie dazu den folgenden Wert für das Attribut **AllowedCipherSpecs** der SSL/TransportSecurity-Zeilengruppe verwenden:

```
AllowedCipher
Specs=TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_128_GCM_S
HA256,TLS_RSA_WITH_AES_256_GCM_SHA384,ECDHE_ECDSA_AES_128_CBC_SHA256,ECDHE_ECD
SA_AES_256_CBC_SHA384,ECDHE_RSA_AES_128_CBC_SHA256,ECDHE_RSA_AES_256_CBC_SHA384,ECDHE_ECD
SA_AES_128_GCM_SHA256,ECDHE_ECDSA_AES_256_GCM_SHA384,ECDHE_RSA_AES_128_GCM_SHA256,ECD
HE_RSA_AES_256_GCM_SHA384
```

Benutzerdefinierte Liste der sortierten und aktivierten CipherSpecs unter IBM MQ for Multiplatforms bereitstellen

Multi

Es ist möglich, dass Sie eine alternative Gruppe von CipherSpecs bereitstellen, die aktiviert sind, und in Ihrer Reihenfolge der Benutzervorgabe für die Verwendung mit IBM MQ-Kanälen entweder die ALW **AMQ_ALLOWED_CIPHERS**-Umgebungsvariable oder das SSL-Zeilengruppenattribut **AllowedCipherSpecs** der Datei `.ini` verwenden. Sie können diese Einstellung aus den folgenden Gründen bevorzugen:

- Um die Annahme eingehender Kanalstartanforderungen von IBM MQ-Listenern zu beschränken, es sei denn, sie verwenden eine der angegebenen CipherSpecs.
- Um die Reihenfolge der in einem TLS-Handshake verwendeten CipherSpecs zu ändern.

Mit dieser Funktion können die CipherSpecs gesteuert werden, die in den CipherSpecs des Typs ANY* enthalten sind.

Die Umgebungsvariable **AMQ_ALLOWED_CIPHERS** oder das Attribut **AllowedCipherSpecs** der SSL-Zeilengruppe akzeptiert:

- Den Namen einer einzelnen CipherSpec.
- Eine durch Kommas getrennte Liste der Namen von CipherSpecs, die erneut aktiviert werden sollen.
- Den Sonderwert ALL, der alle CipherSpecs darstellt.

Anmerkung: Sie sollten nicht **ALL** für die CipherSpecs aktivieren, da dadurch SSL 3.0- und TLS 1.0-Protokolle sowie eine große Zahl schwacher Verschlüsselungsalgorithmen aktiviert werden.

Wenn diese Einstellung konfiguriert ist, überschreibt sie die standardmäßige CipherSpec-Liste und bewirkt, dass IBM MQ die schwachen Einstellungen für die Nichtweiterverwendung der Verschlüsselung ignoriert (siehe unten):

- IBM MQ-Listener akzeptieren nur SSL/TLS-Vorschlägt, die eine der angegebenen CipherSpecs verwenden.
- IBM MQ-Kanäle ermöglichen nur die Verwendung eines leeren SSLCIPH-Werts oder eine der angegebenen CipherSpecs.
- Die Beendigung der Registerkarte **runmqsc** mit SSLCIPH-Werten schränkt die Werte für die Beendigung auf eine der genannten CipherSpecs ein.

Wenn Sie beispielsweise nur zulassen möchten, dass Kanäle definiert/geändert werden und die Zuhörer ECDHE_RSA_AES_128_GCM_SHA256 oder ECDHE_ECDSA_AES_256_GCM_SHA384 akzeptieren, können Sie in der `qm.ini`-Datei Folgendes festlegen:

```
SSL:
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

Mit den CipherSpecs in dieser Liste wird außerdem die Priorität der während eines TLS-Handshakes verwendeten CipherSpecs ermittelt. Wenn Sie beispielsweise eine Liste mit `TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256` angeben, wird beim Handshake wahrscheinlich die CipherSpec TLS_RSA_WITH_AES_128_CBC_SHA256 anstelle der CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256 ausgewählt, wenn ein Client beim Herstellen der Verbindung beide CipherSpecs angibt, d. h. ein Client eine Verbindung mit ANY_TLS12 herstellt.

Beachten Sie, dass von AMQP-oder MQTT-Kanälen verwendete Chiffrierwerte mit den Einstellungen der Datei "java.security" eingeschränkt werden können.

Benutzerdefinierte Liste der sortierten und aktivierten CipherSpecs unter IBM MQ for z/OS bereitstellen

Mithilfe des Zeilengruppenattributs **AllowedCipherSpecs** TransportSecurity des QMINI-Datasetskönnen Sie eine alternative Gruppe von CipherSpecs bereitstellen, die in Ihrer Vorgabenreihenfolge für die Verwendung mit IBM MQ -Kanälen aktiviert sind. Dies kann aus folgenden Gründen gewünscht sein:

- Um die Annahme eingehender Kanalstartanforderungen von IBM MQ-Listenern zu beschränken, es sei denn, sie verwenden eine der angegebenen CipherSpecs.
- Um die Reihenfolge der in einem TLS-Handshake verwendeten CipherSpecs zu ändern.

Mit dieser Funktion können Sie die CipherSpecs steuern, die in den CipherSpecs des Typs ANY* enthalten sind. Das Attribut **AllowedCipherSpecs** akzeptiert:

- Den Namen einer einzelnen CipherSpec.
- Eine durch Kommas getrennte Liste der Namen von CipherSpecs, die erneut aktiviert werden sollen.
- Den Sonderwert ALL, der alle CipherSpecs darstellt.

  Anmerkung: Sie sollten nicht **ALL** für die CipherSpecs aktivieren, da dadurch SSL 3.0- und TLS 1.0-Protokolle sowie eine große Zahl schwacher Verschlüsselungsalgorithmen aktiviert werden. Wenn Sie diese Einstellung konfigurieren, wird die standardmäßige CipherSpec-Liste überschrieben und IBM MQ ignoriert schwache Einstellungen für die Nichtweiterverwendung der Verschlüsselung; siehe „Veraltete CipherSpecs unter z/OS aktivieren” auf Seite 469.

IBM MQ-Listener akzeptieren nur SSL/TLS-Vorschläge, die eine der angegebenen CipherSpecs verwenden, und IBM MQ-Kanäle ermöglichen nur die Verwendung eines leeren SSLCIPH-Werts oder eine der angegebenen CipherSpecs.

Wenn Sie beispielsweise nur zulassen möchten, dass Kanäle geändert oder definiert werden und Listener nur ECDHE_RSA_AES_128_GCM_SHA256 oder ECDHE_RSA_AES_256_GCM_SHA384 akzeptieren, können Sie Folgendes festlegen:

```
TransportSecurity:
AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                   ECDHE_RSA_AES_256_GCM_SHA384
```

Mit den CipherSpecs in dieser Liste wird außerdem die Priorität der während eines TLS-Handshakes verwendeten CipherSpecs ermittelt. Wenn Sie beispielsweise eine Liste mit TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256 angeben, wird beim Handshake wahrscheinlich die CipherSpec TLS_RSA_WITH_AES_128_CBC_SHA256 anstelle der CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256 verwendet, wenn ein Client beim Herstellen der Verbindung beide CipherSpecs angibt, d. h. ein Client eine Verbindung mit ANY_TLS12 herstellt.

Nicht weiter unterstützte CipherSpecs

Eine Liste der veralteten CipherSpecs, die Sie bei Bedarf mit IBM MQ verwenden können.

Anmerkung: Unter AIX, Linux, and Windowsstellt IBM MQ Konformität mit FIPS 140-2 über das Verschlüsselungsmodul "IBM Crypto for C" bereit. Das Zertifikat für dieses Modul wurde in den Langzeitstatus versetzt. Kunden sollten das IBM Crypto for C-Zertifikat anzeigen und sich über Empfehlungen von NIST im Klaren sein. Ein Ersatz-FIPS 140-3-Modul ist derzeit in Bearbeitung und sein Status kann angezeigt werden, indem in der NIST-CMVP-Module in der Prozesslistenach ihm gesucht wird.

Informationen zum Aktivieren veralteter CipherSpecs finden Sie unter „Veraltete CipherSpecs unter IBM MQ for Multiplatforms aktivieren” auf Seite 468 oder „Veraltete CipherSpecs unter z/OS aktivieren” auf Seite 469.

Veraltete CipherSpecs, die Sie mit IBM MQ-TLS-Unterstützung verwenden können, werden in der folgenden Tabelle aufgelistet.

Tabelle 77. Nicht mehr unterstützte CipherSpecs, die Sie für die Verwendung mit IBM MQ wieder aktivieren können

Plattformunterstützung[„1” auf Seite 468]	CipherSpec-Name	Hexadezimalcode	Verwendetes Protokoll	Dateninteg rität	Verschlüsselungsalgorithmus (Verschlüsselungsbits)	FIPS [„2” auf Seite 468]	Suite B	Aktualisierung bei veralteter Version
CipherSpecs für SSL 3.0								
IBM i	AES_SHA_US [„3” auf Seite 468]	002F	SSL 3.0	SHA-1	AES (128)	Nein	Nein	9.0.0.0
Alle	DES_SHA_EXPORT [„3” auf Seite 468] [„4” auf Seite 468] [„5” auf Seite 468]	0009	SSL 3.0	SHA-1	DES (56)	Nein	Nein	9.0.0.0
ALW	DES_SHA_EXPORT1024 [„3” auf Seite 468] [„6” auf Seite 468]	0062	SSL 3.0	SHA-1	DES (56)	Nein	Nein	9.0.0.0
ALW	FIPS_WITH_DES_CBC_SHA [„3” auf Seite 468]	FEFE	SSL 3.0	SHA-1	DES (56)	Nein [„7” auf Seite 468]	Nein	9.0.0.0
ALW	FIPS_WITH_3DES_EDE_CBC_SHA [„3” auf Seite 468]	FEFF	SSL 3.0	SHA-1	3DES (168)	Nein [„8” auf Seite 468]	Nein	9.0.0.1 und 9.0.1
Alle	NULL_MD5 [„3” auf Seite 468]	0001	SSL 3.0	MD5	--	Nein	Nein	9.0.0.1
Alle	NULL_SHA [„3” auf Seite 468]	0002	SSL 3.0	SHA-1	--	Nein	Nein	9.0.0.1
Alle	RC2_MD5_EXPORT [„3” auf Seite 468] [„4” auf Seite 468] [„5” auf Seite 468]	0006	SSL 3.0	MD5	RC2 (40)	Nein	Nein	9.0.0.0
Alle	RC4_MD5_EXPORT [„4” auf Seite 468] [„3” auf Seite 468]	0003	SSL 3.0	MD5	RC4 (40)	Nein	Nein	9.0.0.0

Tabelle 77. Nicht mehr unterstützte CipherSpecs, die Sie für die Verwendung mit IBM MQ wieder aktivieren können (Forts.)

Plattformunterstützung „1" auf Seite 468	CipherSpec-Name	Hexadezimalcode	Verwendetes Protokoll	Datenintegrität	Verschlüsselungsalgorithmus (Verschlüsselungsbits)	FIPS „2" auf Seite 468	Suite B	Aktualisierung bei veralteter Version
Alle	RC4_MD5_US „3" auf Seite 468	0004	SSL 3.0	MD5	RC4 (128)	Nein	Nein	9.0.0.0
Alle	RC4_SHA_US „3" auf Seite 468 „5" auf Seite 468	0005	SSL 3.0	SHA-1	RC4 (128)	Nein	Nein	9.0.0.0
ALW	RC4_56_SHA_EXPORT1024 „3" auf Seite 468 „6" auf Seite 468	0064	SSL 3.0	SHA-1	RC4 (56)	Nein	Nein	9.0.0.0
Alle	TRIPLE_DES_SHA_US „3" auf Seite 468 „5" auf Seite 468	000A	SSL 3.0	SHA-1	3DES (168)	Nein	Nein	9.0.0.1 und 9.0.1
CipherSpecs für TLS 1.0								
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5 „3" auf Seite 468	0006	TLS 1.0	MD5	RC2 (40)	Nein	Nein	9.0.0.0
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5 „3" auf Seite 468 „4" auf Seite 468	0003	TLS 1.0	MD5	RC4 (40)	Nein	Nein	9.0.0.0
Alle	TLS_RSA_WITH_DES_CBC_SHA „3" auf Seite 468	0009	TLS 1.0	SHA-1	DES (56)	Nein „9" auf Seite 468	Nein	9.0.0.0
IBM i	TLS_RSA_WITH_NULL_MD5 „3" auf Seite 468	0001	TLS 1.0	MD5	--	Nein	Nein	9.0.0.1
IBM i	TLS_RSA_WITH_NULL_SHA „3" auf Seite 468	0002	TLS 1.0	SHA-1	--	Nein	Nein	9.0.0.1
IBM i	TLS_RSA_WITH_RC4_128_MD5 „3" auf Seite 468	0004	TLS 1.0	MD5	RC4 (128)	Nein	Nein	9.0.0.0
z/OS ALW	TLS_RSA_WITH_AES_128_CBC_SHA „10" auf Seite 468	002F	TLS 1.0	SHA-1	AES (128)	Ja	Nein	9.0.5
z/OS ALW	TLS_RSA_WITH_AES_256_CBC_SHA „6" auf Seite 468 „10" auf Seite 468	0035	TLS 1.0	SHA-1	AES (256)	Ja	Nein	9.0.5
Alle	TLS_RSA_WITH_3DES_EDE_CBC_SHA	000A	TLS 1.0	SHA-1	3DES (168)	Ja	Nein	9.0.0.1 und 9.0.1
CipherSpecs für TLS 1.2								

Tabelle 77. Nicht mehr unterstützte CipherSpecs, die Sie für die Verwendung mit IBM MQ wieder aktivieren können (Forts.)

Plattformunterstützung „1" auf Seite 468	CipherSpec-Name	Hexadezimalcode	Verwendetes Protokoll	Dateningtegrität	Verschlüsselungsalgorithmus (Verschlüsselungsbits)	FIPS „2" auf Seite 468	Suite B	Aktualisierung bei veralteter Version
ALW	ECDHE_ECDSA_NULL_SHA256 „3" auf Seite 468	C006	TLS 1.2	SHA-1	--	Nein	Nein	9.0.0.1
ALW	ECDHE_ECDSA_RC4_128_SHA256 „3" auf Seite 468	C007	TLS 1.2	SHA-1	RC4 (128)	Nein	Nein	9.0.0.0
ALW IBM i	ECDHE_RSA_NULL_SHA256 „3" auf Seite 468	C010	TLS 1.2	SHA-1	--	Nein	Nein	9.0.0.1
ALW IBM i	ECDHE_RSA_RC4_128_SHA256 „3" auf Seite 468	C011	TLS 1.2	SHA-1	RC4 (128)	Nein	Nein	9.0.0.0
ALW	TLS_RSA_WITH_NULL_NULL „3" auf Seite 468	0000	TLS 1.2	--	--	Nein	Nein	9.0.0.1
Alle	TLS_RSA_WITH_NULL_SHA256 „3" auf Seite 468	003B	TLS 1.2	SHA-256	--	Nein	Nein	9.0.0.1
ALW	TLS_RSA_WITH_RC4_128_SHA256 „3" auf Seite 468	0005	TLS 1.2	SHA-1	RC4 (128)	Nein	Nein	9.0.0.0
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	C0008	TLS 1.2	SHA-1	3DES (168)	Ja	Nein	9.0.0.1 und 9.0.1
ALW IBM i	ECDHE_RSA_3DES_EDE_CBC_SHA256	C012	TLS 1.2	SHA-1	3DES (168)	Ja	Nein	9.0.0.1 und 9.0.1

Tabelle 77. Nicht mehr unterstützte CipherSpecs, die Sie für die Verwendung mit IBM MQ wieder aktivieren können (Forts.)

Plattformunterstützung[1]	CipherSpec-Name	Hexadezimalcode	Verwendetes Protokoll	Dateninteg rität	Verschlüsselungsalgorithmus (Verschlüsselungsbits)	FIPS[2]	Suite B	Aktualisierung bei veralteter Version

Anmerkungen:

1. Eine Liste der Plattformen, die von den einzelnen Plattformsymbolen abgedeckt werden, finden Sie unter Release- und Plattformsymbole in der Produktdokumentation.
2. Gibt an, ob die CipherSpec auf einer FIPS-zertifizierten Plattform FIPS-zertifiziert ist. Unter Federal Information Processing Standards (FIPS) finden Sie eine Beschreibung des FIPS-Standards.
3. ALW Diese CipherSpecs sind inaktiviert, wenn TLS 1.3 aktiviert ist (über die Eigenschaft AllowTLSV13 in der Datei `qm.ini`).

   z/OS Warteschlangenmanager, die mit IBM MQ for z/OS 9.2.0 oder höher erstellt wurden, aktivieren TLS 1.3 standardmäßig, wodurch diese CipherSpecs inaktiviert werden. Sie können diese CipherSpecs bei Bedarf aktivieren, indem Sie TLS V1.3 inaktivieren. Dazu wird **AllowTLSV13**=*FALSE* zur TransportSecurity-Zeilengruppe des QMINI-Datasets in der JCL des Warteschlangenmanagers hinzugefügt. Bei Warteschlangenmanagern, die von einer früheren Version auf IBM MQ for z/OS 9.2.0 migriert wurden, ist TLS 1.3 standardmäßig nicht aktiviert, weshalb diese CipherSpecs aktiviert sind.
4. Die maximale Größe des Handshakeschlüssels beträgt 512 Bit. Hat eines der beim SSL-Handshake ausgetauschten Zertifikate einen Schlüssel mit mehr als 512 Bits, wird ein temporärer 512-Bit-Schlüssel zur Verwendung während des Handshakes generiert.
5. Diese CipherSpecs werden von IBM MQ classes for Java und IBM MQ classes for JMS nicht mehr unterstützt. Weitere Informationen hierzu finden Sie in den Abschnitten SSL/TLS-CipherSpecs und -CipherSuites unter IBM MQ classes for Java und SSL/TLS-CipherSpecs und -CipherSuites unter IBM MQ classes for JMS.
6. Die Größe des Handshakeschlüssels beträgt 1024 Bit.
7. Diese CipherSpec wurde vor dem 19. Mai 2007 FIPS 140-2-zertifiziert. Der Name `FIPS_WITH_DES_CBC_SHA` ist historisch und spiegelt die Tatsache wider, dass diese CipherSpec zuvor FIPS-konform war (aber jetzt nicht mehr). Diese CipherSpec ist veraltet und sollte nicht mehr verwendet werden.
8. Der Name `FIPS_WITH_3DES_EDE_CBC_SHA` ist historisch und spiegelt die Tatsache wider, dass diese CipherSpec zuvor FIPS-konform war (aber jetzt nicht mehr). Die Verwendung dieser CipherSpec wird nicht weiter unterstützt.
9. Diese CipherSpec wurde vor dem 19. Mai 2007 FIPS 140-2-zertifiziert.
10. z/OS Die erneute Aktivierung nur dieser CipherSpecs erfordert nicht die Verwendung der Datendefinitionsanweisung CSQXWEAK.

Veraltete CipherSpecs unter IBM MQ for Multiplatforms aktivieren

Multi

Standardmäßig ist es Ihnen nicht erlaubt, eine veraltete CipherSpec in einer Kanaldefinition anzugeben. Wenn Sie eine veraltete CipherSpec unter IBM MQ for Multiplatforms angeben, erhalten Sie die Nachricht 'AMQ8242: SSLCIPH-Definition falsch' und PCF gibt `MQRCCF_SSL_CIPHER_SPEC_ERROR` zurück.

Sie können keinen Kanal mit einer veralteten CipherSpec starten. Wenn Sie dies mit einer veralteten CipherSpec versuchen, gibt das System `MQCC_FAILED (2)` zusammen mit einem **Reason** von `MQRC_SSL_INITIALIZATION_ERROR (2393)` an den Client zurück.

Sie können eine oder mehrere der veralteten CipherSpecs für die Definition von Kanälen zur Laufzeit auf dem Server wieder aktivieren, indem Sie die Umgebungsvariable **AMQ_SSL_WEAK_CIPHER_ENABLE** festlegen.

Die Umgebungsvariable **AMQ_SSL_WEAK_CIPHER_ENABLE** akzeptiert:

- Ein einzelner CipherSpec-Name oder
- Eine durch Kommas getrennte Liste der Namen von CipherSpecs, die wieder aktiviert werden können, oder
- Den Sonderwert ALL, der alle CipherSpecs darstellt.

Achtung: Bei der Option ALL handelt es sich zwar um eine gültige Option, aber Sie sollten Sie **nur** in bestimmten Situationen verwenden, in denen sie für Ihr Unternehmen erforderlich ist, da bei der erneuten Aktivierung der Option ALL für CipherSpecs die SSL 3.0- und TLS 1.0-Protokolle sowie eine große Zahl schwacher Verschlüsselungsalgorithmen aktiviert werden.

Wenn Sie z. B. ECDHE_RSA_RC4_128_SHA256 erneut aktivieren möchten, legen Sie die folgende Umgebungsvariable fest:

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

Oder ändern Sie alternativ die SSL-Zeilengruppe in der Datei `qm.ini`, indem Sie Folgendes festlegen:

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

Veraltete CipherSpecs unter z/OS aktivieren

z/OS

Standardmäßig ist es Ihnen nicht erlaubt, eine veraltete CipherSpec in einer Kanaldefinition anzugeben. Wenn Sie eine veraltete CipherSpec unter z/OS angeben, erhalten Sie die Nachricht CSQM102E, die Nachricht CSQX616E oder CSQX674E.

Befolgen Sie die in diesem Abschnitt aufgeführten Anweisungen, wenn Sie eine dieser Nachrichten erhalten und Ihr Unternehmen die Verwendung schwacher CipherSpecs erneut aktivieren muss.

Achtung: Damit die Dummy-Definitionsanweisungen (DD) in den folgenden Anweisungen wirksam werden, muss SSLTASKS ein Wert ungleich Null sein. Wenn dies eine Änderung von SSLTASKS erfordert, müssen Sie den Kanalinitiator recyceln.

Unter IBM MQ for z/OS wird zur Steuerung schwacher oder unterbrochener CipherSpecs folgende Methode verwendet:

- Wenn Sie die Verwendung von schwachen CipherSpecs wieder aktivieren möchten, fügen Sie eine Dummy Data Definition (DD)-Anweisung mit dem Namen CSQXWEAKzur JCL des Kanalinitiators hinzu. Wenn diese Option allein angegeben ist, werden nur schwache CipherSpecs aktiviert, die dem TLS 1.2-Protokoll zugeordnet sind. Beispiel:

  ```
  //CSQXWEAK DD DUMMY
  ```

  Anmerkung: Nicht alle veralteten CipherSpecs erfordern die Verwendung dieser DD-Anweisung. Siehe Anmerkung 10 in der vorherigen Tabelle.

- Wenn Sie die Verwendung von SSLv3 -CipherSpecs wieder aktivieren möchten, fügen Sie auch eine Dummy-DD-Anweisung namens CSQXSSL3 zur JCL des Kanalinitiators hinzu. Alle SSLv3 CipherSpecs werden als **Schwach** betrachtet. Daher müssen Sie auch CSQXWEAKangeben:

  ```
  //CSQXSSL3 DD DUMMY
  ```

- Wenn Sie die veralteten TLS V1 -CipherSpecs wieder aktivieren möchten, fügen Sie eine Dummy-DD-Anweisung mit dem Namen TLS10ON (TLS aktivieren V1.0) zur JCL des Kanalinitiators hinzu. Wenn diese Option allein angegeben ist, werden starke CipherSpecs aktiviert, die dem TLS 1.0-Protokoll zugeordnet sind:

  ```
  //TLS10ON DD DUMMY
  ```

  Wenn dies mitCSQXWEAK angegeben wird, werden auch die mit TLS 1.0 verbundenen **Schwachen** CipherSpecs aktiviert.

- Wenn Sie die veralteten TLS- V1 -CipherSpecs explizit inaktivieren möchten, fügen Sie dazu eine Dummy-DD-Anweisung mit dem Namen TLS10OFF (TLS inaktivieren V1.0) zur JCL des Kanalinitiators hinzu. Beispiel:

  ```
  //TLS10OFF DD DUMMY
  ```

Wenn Sie mit dem Listener nur die Verwendung der Verschlüsselungsspezifikationen vereinbaren möchten, die in der Liste **System SSL** mit den Standardverschlüsselungsspezifikationen aufgeführt sind, müssen Sie die folgende DD-Anweisung in der JCL CHINIT definieren:

```
JCL: //GSKDCIPS DD DUMMY
```

Wichtig: V 9.2.0 V 9.2.0 Bei IBM MQ for z/OS 9.2.0 und höher werden die zuvor aufgelisteten DD-Karten und der Wert von **AllowTLSV13** berücksichtigt, wenn Nachrichten während des Kanalinitiatorstarts angezeigt werden, um anzugeben, welche Protokolle aktiviert sind und welche nicht. Selbst wenn eine der zuvor aufgeführten DD-Karten angegeben ist, kann dies also bedeuten, dass aufgrund einer Kombination dieser Einstellungen ein bestimmten Protokoll nicht mit einem anderen Protokoll aktiviert werden kann. Beispielsweise ist das Protokoll SSL 3.0 nicht zulässig, wenn TLS 1.3 aktiviert ist.

Es gibt alternative Mechanismen, die verwendet werden können, um schwache CipherSpecs und SSLv3-Unterstützung zwangsweise erneut zu aktivieren, falls die Änderung der Datendefinition nicht geeignet ist. Wenden Sie sich für weitere Informationen an den IBM Service.

Zugehörige Konzepte

„Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ” auf Seite 49
Dieser Abschnitt enthält Informationen dazu, wie Sie die richtigen CipherSpecs und digitalen Zertifikate für Ihre Sicherheitsrichtlinie auswählen. Dazu wird die Beziehung zwischen CipherSpecs und digitalen Zertifikaten in IBM MQ erläutert.

Zugehörige Verweise

CHANNEL DEFINE CHANNEL
ALTER CHANNEL

Beziehung zwischen Einstellungen für Alias-CipherSpecs

In diesen Informationen wird das erwartete Verhalten mit unterschiedlichen Kombinationen aus Alias-CipherSpecs in Client- und Serverkonfigurationen beschreiben. Dabei bezeichnet ein Client die Entität, die die Kommunikation einleitet, beispielsweise eine Clientanwendung oder ein Senderkanal eines Warteschlangenmanagers, und Server bezeichnet die Entität, die die Kommunikation vom Client empfängt, beispielsweise ein Serververbindungskanal oder ein Empfängerkanal.

CipherSpecs mit Mindestprotokoll im Vergleich zu festgelegtem Protokoll

V 9.2.0

IBM MQ unterstützt zwei verschiedene Typen von CipherSpecs:

Mindestprotokoll
Bei CipherSpecs mit einem Mindestprotokoll wird keine Obergrenze festgelegt, z. B. ANY, ANY_TLS12_OR_HIGHER oder ANY_TLS13_OR_HIGHER.

Festgelegtes Protokoll
Bei CipherSpecs mit einem festgelegten Protokoll wird ein bestimmtes Protokoll angegeben, z. B. ANY_TLS12 und ANY_TLS13, oder es wird ein bestimmter Algorithmus angegeben, z. B. ECDHE_ECDSA_3DES_EDE_CBC_SHA256.

Ab IBM MQ 9.2.0 werden CipherSpecs mit Mindestprotokoll und festgelegtem Protokoll auf allen Plattformen unterstützt.

Um die Einfachheit der Konfiguration zu maximieren und gleichzeitig die Sicherheit zu gewährleisten, empfiehlt es sich, die Verwendung von CipherSpecs mit dem **Mindestprotokoll** auf beiden Seiten des Kanals zu verwenden. Dies ermöglicht es Ihrer Kommunikation, eine höhere TLS-Protokollversion automatisch zu unterstützen und zu verwenden, wenn beide Seiten eine neue Version unterstützen, ohne dass eine Änderung der Seitenkonfiguration erforderlich ist.

Wenn Sie eine CipherSpec mit einem **Mindestprotokoll** auf der einleitenden Seite, aber eine CipherSpec mit einem **festgelegten Protokoll** auf der Empfangsseite verwenden, kann dies dazu führen, dass die Verbindung zurückgegeben wird und die

- Multi Nachrichten AMQ9631 und AMQ9641 ausgegeben werden.
- V 9.2.0 z/OS V 9.2.0 Nachrichten CSQX631E und CSQX641E ausgegeben werden.

In der folgenden Tabelle wird die Beziehung zwischen den Einstellungen der Alias-CipherSpec und dem erwarteten Ergebnis gezeigt. Tabelle 78 auf Seite 471 zeigt das erwartete Verhalten, wenn TLS 1.3 auf dem Client und/oder dem Server nicht aktiviert ist. Tabelle 79 auf Seite 472 zeigt das erwartete Verhalten, wenn TLS 1.3 sowohl auf dem Client als auch auf dem Server aktiviert ist. In beiden Fällen werden die CipherSpecs für den Client auf der Y-Achse der Tabelle und die CipherSpecs für den Server auf der X-Achse der Tabelle angezeigt.

Anmerkung: In den folgenden Tabellen zeigen die Zellen mit der Markierung *Schlägt wahrscheinlich fehl* das Potenzial für einen Konflikt an, wenn Sie eine CipherSpec **minimum protocol** für einen Teil einer Verbindung angeben, und ein bestimmtes (**festes Protokoll**) CipherSpec für einen anderen Teil.

Angenommen, der Client und der Server sind so festgelegt, dass sie ein beliebiges CipherSpec verwenden, und der Serverkanal wird für die Verwendung eines bestimmten CipherSpec festgelegt:

- Wenn das stärkste unterstützte CipherSpec sowohl für den Client als auch für den Server mit dem spezifischen CipherSpec übereinstimmt, das auf dem Kanal konfiguriert ist, wird der TLS-Handshake erfolgreich aufgelöst.
- Wenn jedoch ein stärkeres CipherSpec vorhanden ist, das sowohl die Client- als auch die Serverunterstützung unterstützt, wird der TLS-Handshake so aufgelöst, dass er dies verwendet, auch wenn er nicht mit dem auf dem Kanal angegebenen CipherSpec übereinstimmt und der TLS-Handshake fehlschlägt.

Tabelle 78. Erwartetes Verhalten, wenn TLS 1.3 auf dem Client und/oder dem Server nicht aktiviert ist

	Server			
Client	**Spezifische TLS 1.2-CipherSpec**	**ANY**	**ANY_ TLS12**	**ANY_TLS12_ OR_HIGHER**
Spezifische TLS 1.2-CipherSpec	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen
Beliebig	*Schlägt wahrscheinlich fehl*	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen
ANY_ TLS12	*Schlägt wahrscheinlich fehl*	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen

Tabelle 78. Erwartetes Verhalten, wenn TLS 1.3 auf dem Client und/oder dem Server nicht aktiviert ist (Forts.)				
	Server			
Client	**Spezifische TLS 1.2-CipherSpec**	**ANY**	**ANY_ TLS12**	**ANY_TLS12_ OR_HIGHER**
ANY_TLS12_ OR_HIGHER	*Schlägt wahrscheinlich fehl*	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen

Tabelle 79. Erwartetes Verhalten, wenn TLS 1.3 sowohl auf dem Client als auch auf dem Server aktiviert ist.							
	Server						
Client	**Spezifische TLS 1.2-CipherSpec**	**Spezifische TLS 1.3-CipherSpec**	**ANY**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ OR_HIGHER**	**ANY_TLS13_ OR_HIGHER**
Spezifische TLS 1.2-CipherSpec	Verbindungsherstellungen	**Schlägt fehl**	Verbindungsherstellungen	Verbindungsherstellungen	**Schlägt fehl**	Verbindungsherstellungen	**Schlägt fehl**
Spezifische TLS 1.3-CipherSpec	**Schlägt fehl**	Verbindungsherstellungen	Verbindungsherstellungen	**Schlägt fehl**	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen
Beliebig	**Schlägt fehl**	*Schlägt wahrscheinlich fehl*	Verbindungsherstellungen	**Schlägt fehl**	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen
ANY_TLS12	*Schlägt wahrscheinlich fehl*	**Schlägt fehl**	Verbindungsherstellungen	Verbindungsherstellungen	**Schlägt fehl**	Verbindungsherstellungen	**Schlägt fehl**
ANY_TLS13	**Schlägt fehl**	*Schlägt wahrscheinlich fehl*	Verbindungsherstellungen	**Schlägt fehl**	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen
ANY_TLS12_ OR_HIGHER	**Schlägt fehl**	*Schlägt wahrscheinlich fehl*	Verbindungsherstellungen	**Schlägt fehl**	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen
ANY_TLS13_ OR_HIGHER	**Schlägt fehl**	*Schlägt wahrscheinlich fehl*	Verbindungsherstellungen	**Schlägt fehl**	Verbindungsherstellungen	Verbindungsherstellungen	Verbindungsherstellungen

Zugehörige Konzepte

„Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ” auf Seite 49
Dieser Abschnitt enthält Informationen dazu, wie Sie die richtigen CipherSpecs und digitalen Zertifikate für Ihre Sicherheitsrichtlinie auswählen. Dazu wird die Beziehung zwischen CipherSpecs und digitalen Zertifikaten in IBM MQ erläutert.

„CipherSpecs und CipherSuites” auf Seite 21
Kryptografische Sicherheitsprotokolle müssen sich auf die Algorithmen einigen, die von einer sicheren Verbindung verwendet werden. CipherSpecs und CipherSuites definieren bestimmte Kombinationen von Algorithmen.

„CipherSpecs aktivieren” auf Seite 448
Aktivieren Sie eine CipherSpec mit dem Parameter **SSLCIPH** im MQSC-Befehl **DEFINE CHANNEL** oder **ALTER CHANNEL**.

Zugehörige Tasks

Vorhandene Sicherheitskonfigurationen für die Verwendung der CipherSpec ANY_TLS12_OR_HIGHER migrieren

Informationen zu CipherSpecs mithilfe von IBM MQ Explorer anfordern

Sie können IBM MQ Explorer verwenden, um Beschreibungen von CipherSpecs anzuzeigen.

Gehen Sie wie folgt vor, um Informationen zu den CipherSpecs in „CipherSpecs aktivieren” auf Seite 448 abzurufen:

1. Öffnen Sie IBM MQ Explorer und erweitern Sie den Ordner **Warteschlangenmanager**.
2. Stellen Sie sicher, dass der WS-Manager gestartet wurde.
3. Wählen Sie den Queue Manager aus, mit dem Sie arbeiten möchten, und klicken Sie auf **Kanäle**.
4. Klicken Sie auf den Kanal, mit dem Sie arbeiten wollen, und wählen Sie **Eigenschaften** aus.
5. Wählen Sie die Eigenschaftenseite **SSL** aus.
6. Wählen Sie in der Liste die CipherSpec aus, mit der gearbeitet werden soll. Eine Beschreibung wird im Fenster unterhalb der Liste angezeigt.

z/OS IBM i Alternativen für die Angabe von CipherSpecs

Auf Plattformen, auf denen die TLS-Unterstützung vom Betriebssystem zur Verfügung gestellt wird, werden eventuell auch neue CipherSpecs unterstützt, die nicht in „CipherSpecs aktivieren” auf Seite 448 enthalten sind.

Sie können eine neue CipherSpec mit dem Parameter SSLCIPH angeben, aber der von Ihnen angegebene Wert hängt von Ihrer Plattform ab. In allen Fällen muss die Spezifikation einer TLS-CipherSpec entsprechen, die sowohl gültig als auch von der Version von TLS unterstützt wird, auf der Ihr System ausgeführt wird.

Anmerkung: Dieser Abschnitt gilt nicht für AIX, Linux, and Windows-Systeme, da die CipherSpecs mit dem IBM MQ-Produkt bereitgestellt werden, sodass neue CipherSpecs nach dem Versand nicht verfügbar werden.

IBM i IBM i

Eine Zeichenfolge mit zwei Zeichen, die einen Hexadezimalwert darstellt.

Weitere Informationen zu den zulässigen Werten finden Sie unter Punkt 3 im Abschnitt "Hinweise zur Verwendung" von Zeicheninformationen für eine sichere Sitzung festlegen.

Achtung: Sie sollten keine hexadezimalen Cipher-Werte in **SSLCIPH** angeben, da aus dem Wert, der Chiffrierwert verwendet wird, unklar ist, und die Auswahl des zu verwendenden Protokolls unbestimmt ist. Die Verwendung von hexadezimalen Chiffrierungswerten kann zu Fehlern bei CipherSpec-Fehlern führen.

Sie können den Wert mit dem Befehl **CHGMQMCHL** oder **CRTMQMCHL** angeben; Beispiel:

```
CRTMQMCHL CHLNAME(' channel name ') SSLCIPH(' hexadecimal value ')
```

Sie können auch den MQSC-Befehl **ALTER QMGR** verwenden, um den Parameter **SSLCIPH** festzulegen.

z/OS z/OS

Eine vierstellige Zeichenfolge, die einen Hexadezimalwert darstellt. Die hexadezimalen Codes entsprechen den Werten, die im TLS-Protokoll definiert sind.

Weitere Informationen finden Sie in den Cipher-Suite-Definitionen, in denen eine Liste aller unterstützten Verschlüsselungsspezifikationen für TLS 1.0, TLS 1.2 und TLS 1.3 in Form von vierstelligen hexadezimalen Codes enthalten ist.

Anmerkung: Um eine schwache CipherSpec oder eine CipherSpec, die zu einem veralteten Protokoll gehört, wie SSL V3.0 oder TLS 1.0, verwenden zu können, müssen Sie die entsprechende DD-Karte in der Start-JCL des Kanalinitiators angeben. Weitere Informationen finden Sie im Abschnitt „Nicht weiter unterstützte CipherSpecs" auf Seite 465.

Hinweise zu IBM MQ-Clustern

Für IBM MQ-Cluster sollten Sie die CipherSpec-Namen verwenden, die unter „CipherSpecs aktivieren" auf Seite 448 angegeben werden. Wenn Sie eine alternative Spezifikation verwenden, müssen Sie beachten, dass die Spezifikation auf anderen Plattformen möglicherweise nicht gültig ist. Weitere Informationen hierzu finden Sie unter „SSL/TLS und Cluster" auf Seite 515.

CipherSpec für einen IBM MQ MQI client angeben

Sie haben drei Optionen für die Angabe eines CipherSpec für einen IBM MQ MQI client.

Diese Optionen lauten wie folgt:

- Verwenden einer Kanaldefinitionstabelle
- Verwenden Sie das Feld SSLCipherSpec in der MQCD-Struktur, in MQCD_VERSION_7 oder höher oder in einem MQCONNX-Aufruf.
- Verwendung von Active Directory (auf Windows-Systemen mit Active Directory-Unterstützung)

CipherSuite mit IBM MQ classes for Java und IBM MQ classes for JMS angeben

In IBM MQ classes for Java und IBM MQ classes for JMS werden CipherSuites anders als auf anderen Plattformen angegeben.

Weitere Informationen zur Angabe einer CipherSuite mit IBM MQ classes for Java finden Sie unter Unterstützung von Transport Layer Security (TLS) für Java

Weitere Informationen zur Angabe einer CipherSuite mit IBM MQ classes for JMS finden Sie unter Transport Layer Security (TLS) mit IBM MQ classes for JMS verwenden.

CipherSpec für IBM MQ.NET angeben

Für IBM MQ.NET können Sie die CipherSpec mit der MQEnvironment-Klasse oder unter Verwendung von MQC.SSL_CIPHER_SPEC_PROPERTY in der Hashtabelle der Verbindungseigenschaften angeben.

Weitere Informationen zur Angabe einer CipherSpec für den nicht verwalteten .NET-Client finden Sie im Abschnitt TLS für den nicht verwalteten .NET-Client aktivieren

Weitere Informationen zur Angabe einer CipherSpec für den verwalteten .NET-Client finden Sie unter CipherSpec-Unterstützung für den verwalteten .NET-Client

z/OS Verwendung von AT-TLS mit IBM MQ for z/OS

Application Transparent Transport Layer Security (AT-TLS) stellt TLS-Unterstützung für Anwendungen von z/OS bereit, ohne dass Anwendungen TLS-Unterstützung implementieren müssen, oder sogar darauf achten, dass TLS verwendet wird. AT-TLS ist nur unter z/OS verfügbar.

AT-TLS kann mit allen Versionen von IBM MQ for z/OS verwendet werden.

Stellen Sie vor der Verwendung von AT-TLS mit IBM MQ for z/OS sicher, dass Sie die beteiligten „Einschränkungen" auf Seite 478 verstehen.

Für die Verwendung von Application Transparent Transport Layer Security definieren Sie Richtlinienanweisungen, die eine Gruppe von Regeln enthalten, die von z/OS Communications Server verwendet werden, um zu entscheiden, welche TCP/IP-Verbindungen TLS transparent aktiviert haben.

IBM MQ for z/OS verfügt über eine eigene TLS-Implementierung, die erfordert, dass Kanäle den SSLCIPH-Parameter mit einer unterstützten CipherSpec konfiguriert haben.

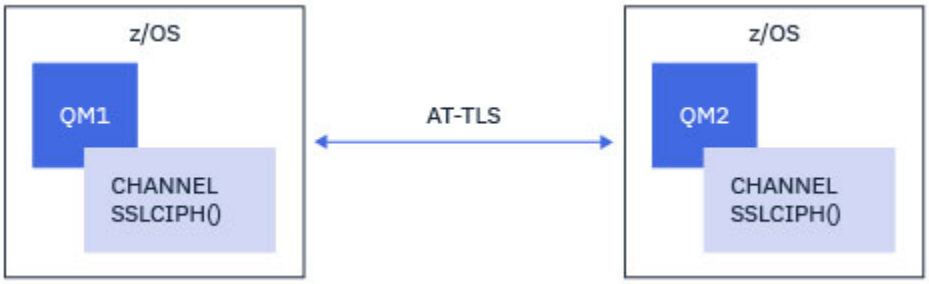
Bei der Entscheidung, TLS auf einem Kanal zu aktivieren, kann der IBM MQ-Administrator entscheiden, ob er AT-TLS oder IBM MQ TLS verwenden möchte. Die Entscheidung basiert häufig darauf, ob AT-TLS für andere Middleware verwendet wird, oder wird aufgrund von Auswirkungen auf die Leistung getroffen. Für einen grundlegenden Vergleich der Leistung von AT-TLS und IBM MQ TLS siehe MP16: Kapazitätsplanung und -optimierung für IBM MQ for z/OS.

Szenarien

Die Verwendung von AT-TLS mit IBM MQ wird in den folgenden Szenarios unterstützt:

Szenario 1

Zwischen zwei IBM MQ for z/OS-Queue Managern, bei denen beide Seiten des Kanals AT-TLS verwenden. Das heißt, kein Kanal gibt das Attribut SSLCIPH an. Dieser Ansatz kann mit jedem Nachrichtenkanal verwendet werden.



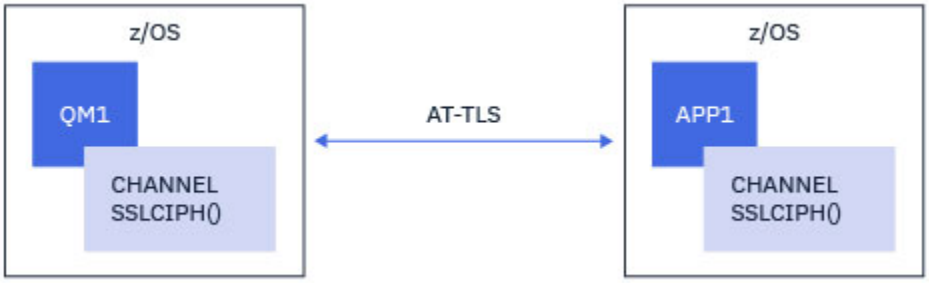
Die Implementierung dieses Szenarios besteht darin, zwei AT-TLS-Richtlinien zu definieren, eine für jede Seite des Kanals. Diese Richtlinien sind mit den Richtlinien identisch, die in Szenario 3 oder Szenario 4 verwendet werden.

Wenn beispielsweise der Kanal von der Verwendung einer einzelnen, benannten CipherSpec zur Verwendung von AT-TLS geändert wurde, würde der abgehende Kanal die Richtlinie von „Konfigurieren von AT-TLS auf einem Kanal für abgehende Nachrichten zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec" auf Seite 479 verwenden und der eingehende Kanal würde die Richtlinie von „Konfigurieren von AT-TLS in einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec" auf Seite 488 verwenden.

Wenn der Kanal von der Verwendung einer Alias-CipherSpec zur Verwendung von AT-TLS geändert wurde, würde der abgehende Kanal die Richtlinie von „Konfigurieren von AT-TLS auf einem abgehenden Kanal zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung von Alias-CipherSpecs" auf Seite 483 verwenden und der eingehende Kanal würde die Richtlinie von „Konfigurieren von AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer Alias-CipherSpec" auf Seite 493 verwenden.

Szenario 2

Zwischen einem IBM MQ for z/OS-Queue Manager und einer IBM MQ Java-Client-Anwendung, die unter z/OS ausgeführt wird, wobei beide Seiten des Kanals AT-TLS verwenden. Das bedeutet, dass weder der Serververbindungskanal noch der Clientverbindungskanal das Attribut SSLCIPH angeben.



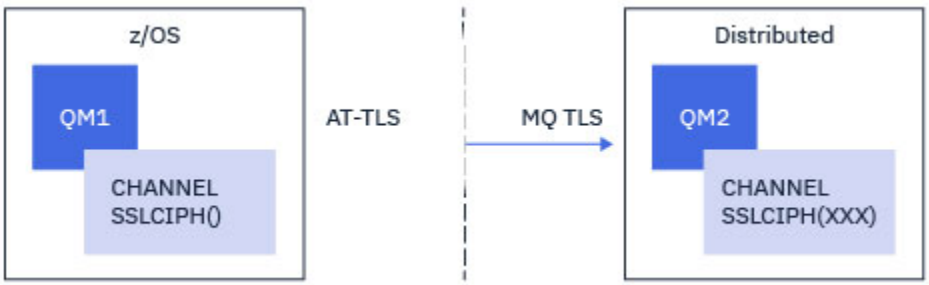
Die Implementierung dieses Szenarios besteht darin, zwei AT-TLS-Richtlinien zu definieren, eine für jede Seite des Kanals. Diese Richtlinien sind mit den Richtlinien identisch, die in Szenario 3 oder Szenario 4 verwendet werden.

Wenn beispielsweise der Kanal von der Verwendung einer einzigen, benannten CipherSpec zur Verwendung von AT-TLS geändert wurde, würde der Clientverbindungskanal die Richtlinie von „Konfigurieren von AT-TLS auf einem Kanal für abgehende Nachrichten zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec" auf Seite 479 verwenden, und der Serververbindungskanal würde die Richtlinie von „Konfigurieren von AT-TLS in einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec" auf Seite 488 verwenden.

Wenn der Kanal von der Verwendung einer AliasCipherSpec für die Verwendung von AT-TLS geändert wurde, würde der Clientverbindungskanal die Richtlinie von „Konfigurieren von AT-TLS auf einem abgehenden Kanal zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung von Alias-CipherSpecs" auf Seite 483 verwenden und der Serververbindungskanal würde die Richtlinie von „Konfigurieren von AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer Alias-CipherSpec" auf Seite 493 verwenden.

Szenario 3

Zwischen einem IBM MQ for z/OS-Queue Manager und einem Queue Manager, der unter IBM MQ for Multiplatforms ausgeführt wird, wobei der IBM MQ for z/OS-Queue Manager AT-TLS verwendet und der IBM MQ for Multiplatforms-Queue Manager IBM MQ TLS verwendet. Dies gilt für alle Nachrichtenkanaltypen außer Cluster-Sender und Cluster-Empfänger.

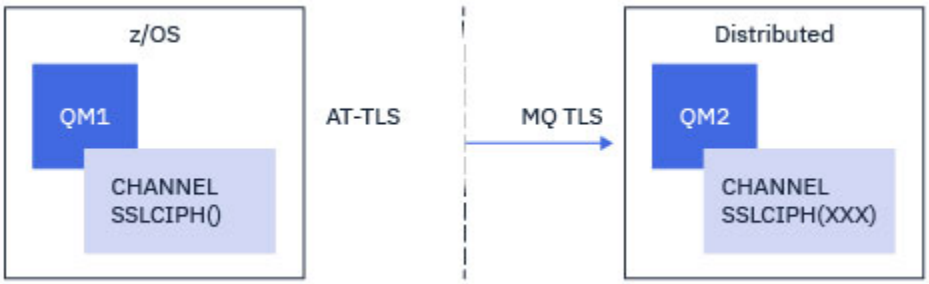


In „Konfigurieren von AT-TLS auf einem Kanal für abgehende Nachrichten zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec" auf Seite 479 finden Sie eine Beispiel-AT-TLS-Konfiguration für abgehende Kanäle vom IBM MQ for z/OS-Queue Manager zum IBM MQ for Multiplatforms-Queue Manager und in „Konfigurieren von AT-TLS in einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec" auf Seite 488 für eine Beispiel-AT-TLS-Konfiguration für eingehende Kanäle vom IBM MQ for Multiplatforms-Queue Manager zum IBM MQ for z/OS-Queue Manager.

Die gleiche AT-TLS-Konfiguration kann verwendet werden, wenn sich beide Queue Manager auf z/OS befinden, der Queue Manager auf der rechten Seite jedoch nicht für die Verwendung von AT-TLS konfiguriert wurde.

Szenario 4

Zwischen einem IBM MQ for z/OS-Queue Manager und einem Queue Manager, der unter IBM MQ for Multiplatforms ausgeführt wird, wobei der IBM MQ for z/OS-Queue Manager AT-TLS verwendet und der IBM MQ for Multiplatforms-Queue Manager IBM MQ TLS verwendet, indem er das Attribut SSLCIPH mit einer Alias-CipherSpec angibt. Dies gilt für alle Nachrichtenkanaltypen außer Cluster-Sender und Cluster-Empfänger.

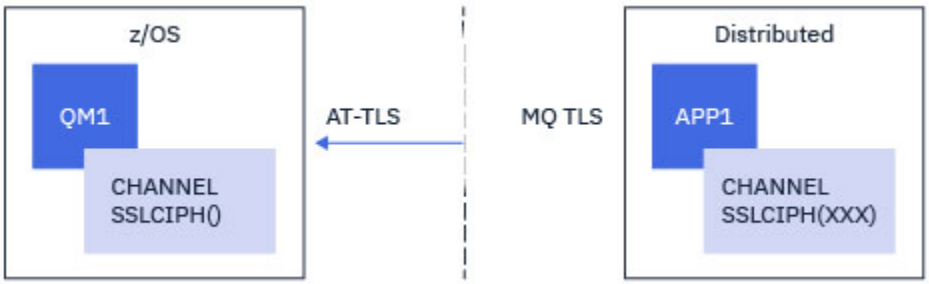


In „Konfigurieren von AT-TLS auf einem abgehenden Kanal zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung von Alias-CipherSpecs" auf Seite 483 finden Sie eine Beispiel-AT-TLS-Konfiguration für abgehende Kanäle vom IBM MQ for z/OS-Queue Manager zum IBM MQ for Multiplatforms-Queue Manager sowie „Konfigurieren von AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer Alias-CipherSpec" auf Seite 493 und „Konfigurieren von AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer Alias-CipherSpec" auf Seite 493 für eine Beispiel-AT-TLS-Konfiguration für eingehende Kanäle vom IBM MQ for Multiplatforms-Queue Manager zum IBM MQ for z/OS-Queue Manager.

Die gleiche AT-TLS-Konfiguration kann verwendet werden, wenn sich beide Queue Manager auf z/OS befinden, der Queue Manager auf der rechten Seite jedoch nicht für die Verwendung von AT-TLS konfiguriert wurde.

Szenario 5

Zwischen einem IBM MQ for z/OS-Queue Manager und einer Clientanwendung, die unter IBM MQ for Multiplatforms ausgeführt wird, wobei der IBM MQ for z/OS-Queue Manager AT-TLS verwendet und die Clientanwendung IBM MQ TLS verwendet, indem das Attribut SSLCIPH mit einer einzigen, benannten CipherSpec angegeben wird.

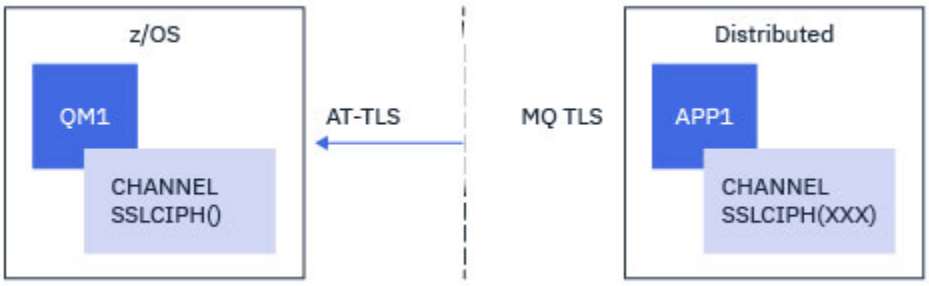


Für dieses Szenario ist eine einzelne AT-TLS-Richtlinie erforderlich, die dieselben Anforderungen erfüllt wie die von einem eingehenden Nachrichtenkanal. Informationen hierzu finden Sie im Artikel „Konfigurieren von AT-TLS in einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec" auf Seite 488.

Die gleiche AT-TLS-Konfiguration kann verwendet werden, wenn die Clientanwendung eine Java-Anwendung ist und auch unter z/OS ausgeführt wird, aber nicht für die Verwendung von AT-TLS konfiguriert wurde.

Szenario 6

Zwischen einem IBM MQ for z/OS-Queue Manager und einer Clientanwendung, die unter IBM MQ for Multiplatforms ausgeführt wird, wobei der IBM MQ for z/OS-Queue Manager AT-TLS verwendet und die Clientanwendung IBM MQ TLS verwendet, indem das Attribut SSLCIPH mit einer Alias-CipherSpec angegeben wird.



Für dieses Szenario ist eine einzelne AT-TLS-Richtlinie erforderlich, die dieselben Anforderungen erfüllt wie die von einem eingehenden Nachrichtenkanal. Informationen hierzu finden Sie im Artikel „Konfigurieren von AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer Alias-CipherSpec” auf Seite 493.

Die gleiche AT-TLS-Konfiguration kann verwendet werden, wenn die Clientanwendung eine Java-Anwendung ist und auch unter z/OS ausgeführt wird, aber nicht für die Verwendung von AT-TLS konfiguriert wurde.

Einschränkungen

IBM MQ for z/OS ist sich nicht über AT-TLS bewusst, daher gibt es mehrere Einschränkungen, die mit den vorherigen Szenarios gelten:

- AT-TLS in Kombination mit IBM MQ TLS funktioniert nicht mit Cluster-Sender-und Clusterempfängerkanälen.
- IBM MQ for z/OS-Queue Manager sind sich nicht bewusst, dass sie AT-TLS verwenden und keine Zertifikatsinformationen von ihrem Partnerwarteschlangenmanager oder Client empfangen. Aus diesem Grund haben die folgenden Attribute keine Auswirkung auf die z/OS-Seite eines Kanals, der AT-TLS verwendet:
  - Die Attribute "SSLCAUTH" und "SSLPEER"
  - SSLRKEYC-Warteschlangenmanagerattribut
  - Die SSLPEERMAP-Attribute von CHLAUTH-Regeln
- Für die Verwendung der geheimen TLS-Schlüsselneuvereinbarung ist es erforderlich, dass beide Seiten des Kanals IBM MQ TLS verwenden. Aus diesem Grund sollte bei einem IBM MQ for Multiplatforms-Queue Manager oder -Client die Neuvereinbarung der geheimen TLS-Schlüssel nicht aktiviert sein, wenn eine Verbindung zu einem IBM MQ for z/OS-Queue Manager mit AT-TLS hergestellt wird.

  Um die Neuaushandlung des geheimen TLS-Schlüssels für einen Warteschlangenmanager zu inaktivieren, setzen Sie den Parameter SSLRKEYC des Warteschlangenmanagers auf `0`. Setzen Sie für einen Client den relevanten Parameter je nach Clienttyp auf `0` . Details zur Vorgehensweise finden Sie unter „Zurücksetzen von geheimen SSL-und TLS-Schlüsseln” auf Seite 497.

AT-TLS-Konfigurationsanweisungen

AT-TLS wird mit einer Gruppe von Anweisungen konfiguriert. In den Szenarios, die in diesem Abschnitt dokumentiert sind, werden folgende verwendet:

TTLSRule
: Gibt eine Gruppe von Kriterien für die Anpassung einer TCP/IP-Verbindung an eine TLS-Konfiguration an. Dies wiederum bezieht sich auf die anderen Anweisungstypen.

TTLSGroupAction
Gibt an, ob die Referenzierung `TTLSRule` aktiviert ist oder nicht.

TTLSEnvironmentAction
Gibt die detaillierte Konfiguration für die Referenzierung `TTLSRule` an und verweist auf eine Reihe anderer Anweisungen.

TTLSKeyringParms
Verweist auf den Schlüsselring, der von AT-TLS verwendet werden soll.

TTLSCipherParms
Definiert die Cipher Suites, die verwendet werden sollen.

TTLSEnvironmentAdvancedParms
Definiert, welche TLS-oder SSL-Protokolle aktiviert sind.

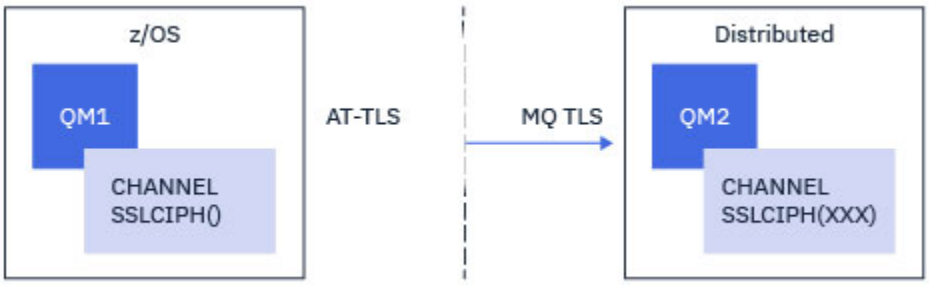
Achtung: Es gibt andere AT-TLS-Richtlinienanweisungen mit AT-TLS, die hier nicht dokumentiert sind, und kann je nach Bedarf mit IBM MQ verwendet werden. IBM MQ wurde jedoch nur mit den Richtlinien getestet, die in diesem Abschnitt beschrieben sind.

Konfigurieren von AT-TLS auf einem Kanal für abgehende Nachrichten zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec

Wie Sie AT-TLS auf einem abgehenden Kanal von einem IBM MQ for z/OS-Queue Manager auf einen IBM MQ for Multiplatforms-Queue Manager einrichten. In diesem Fall ist der Kanal auf dem z/OS-Queue Manager ein Senderkanal, der nicht über das Attribut SSLCIPH verfügt, und der Kanal auf dem Queue Manager ohne z/OS ist ein Empfängerkanal, dessen Attribut SSLCIPH auf eine einzige benannte CipherSpec gesetzt ist.

Ein Beispiel für ein Beispiel unter Verwendung einer AliasCipherSpec finden Sie in „Konfigurieren von AT-TLS auf einem abgehenden Kanal zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung von Alias-CipherSpecs” auf Seite 483.

In diesem Beispiel wird ein vorhandenes Sender-Empfänger-Kanalpaar, das die ANY_TLS13-Alias-CipherSpec verwendet, so angepasst, dass der Senderkanal AT-TLS anstelle von IBM MQ TLS verwendet.



In diesem Beispiel wird ein vorhandenes Sender-Empfänger-Kanalpaar, das die TLS 1.3 TLS_AES_256_GCM_SHA384-CipherSpec verwendet, so angepasst, dass der Senderkanal AT-TLS anstelle von IBM MQ TLS verwendet.

Andere TLS-Protokolle und CipherSpecs können verwendet werden, indem kleinere Anpassungen an der Konfiguration vorgenommen werden. Andere Nachrichtenkanaltypen können, abgesehen von Clustersender- und Clusterempfängerkanälen, ohne Änderung der AT-TLS-Konfiguration verwendet werden.

Achtung: TLS 1.3 kann nur in z/OS Version 2.4 oder höher verwendet werden.

Verfahren

Schritt 1: Kanal stoppen

Schritt 2: Eine AT-TLS-Richtlinie erstellen und anwenden

Für dieses Szenario müssen Sie die folgenden AT-TLS-Anweisungen erstellen:

1. Eine Anweisung TTLSRule, mit der abgehende Verbindungen vom Kanalinitiatoradressbereich an die IP-Adresse und die Portnummer des Zielempfängerkanals abgeglichen werden sollen. Diese Werte sollten mit den Informationen übereinstimmen, die in CONNAME des Senderkanals verwendet werden. Hier wurde eine weitere Filterung eingeschlossen, die einem bestimmten Kanalinitiatorjobnamen entspricht.

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

   Die vorherige Regel stimmt mit Verbindungen überein, die mit der IP-Adresse 123.456.78.9 an Port 1414 aus dem Job CSQ1CHIN gehen.

   Weitere erweiterte Filteroptionen werden unter TTLSRule beschrieben.

2. Eine Anweisung TTLSGroupAction, mit der die Regel aktiviert wird. Der TTLSRule verweist auf die TTLSGroupAction mit der Eigenschaft **TTLSGroupActionRef**.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Eine TTLSEnvironmentAction-Anweisung, die der TTLSRule durch die Eigenschaft **TTLSEnvironmentActionRef** zugeordnet ist. Ein TTLSEnvironmentAction konfiguriert die TLS-Umgebung und gibt an, welcher Schlüsselring verwendet werden soll.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. Eine TTLSKeyringParms-Anweisung, die der TTLSEnvironmentAction durch die Eigenschaft **TTLSKeyringParmsRef** zugeordnet ist und den Schlüsselring definiert, der von AT-TLS verwendet wird.

   Der Schlüsselring sollte Zertifikate enthalten, die von dem remote Nicht-z/OS-Queue Manager anerkannt werden. Dieser Schlüsselring kann auf die gleiche Weise wie ein Schlüsselring definiert werden, der vom Kanalinitiator verwendet wird (siehe „z/OS-System für die Verwendung von TLS konfigurieren” auf Seite 274).

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Eine TTLSCipherParms-Anweisung, die der TTLSEnvironmentAction durch die Eigenschaft **TTLSCipherParmsRef** zugeordnet ist.

Diese Anweisung muss einen einzelnen Cipher-Suite-Namen enthalten, der dem auf dem Zielempfängerkanal verwendeten IBM MQ-CipherSpec-Namen entsprechen muss.

Anmerkung: AT-TLS-Cipher-Suite-Namen stimmen nicht unbedingt mit IBM MQ-CipherSpec-Namen überein. Es ist jedoch möglich, den AT-TLS-Cipher-Suite-Namen zu finden, der mit einem IBM MQ-CipherSpec-Namen übereinstimmt, indem der IBM MQ-CipherSpec-Name in der folgenden Tabelle gesucht wird und die hexadezimale Codespalte mit der erweiterten Zeichenspalte aus Tabelle 2 in der TTLSCipherParms-Anweisungsgruppe referenziert wird.

Tabelle 80. CipherSpecs auf z/OS aus IBM MQ for z/OS 9.2.0

CipherSpec	Protokoll	Hexadezimalcode	Standardmäßig aktiviert
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Ja
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Ja
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Ja
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Ja
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Ja
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Ja
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Ja
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Ja
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Ja
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Ja
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Ja
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Ja
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Ja
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Nein
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Nein

Tabelle 80. CipherSpecs auf z/OS aus IBM MQ for z/OS 9.2.0 (Forts.)			
CipherSpec	**Protokoll**	**Hexadezimalcode**	**Standardmäßig aktiviert**
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Nein
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Nein
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Nein
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Nein
TRIPLE_DES_SHA_US	SSL v3	000A	Nein
RC4_SHA_US	SSL v3	0005	Nein
RC4_MD5_US	SSL v3	0004	Nein
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Nein
RC2_MD5_EXPORT	SSL v3	0006	Nein
NULL_SHA	SSL v3	0002	Nein
NULL_MD5	SSL v3	0001	Nein

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. Eine Anweisung TTLSEnvironmentAdvancedParms wird der TTLSEnvironmentAction durch die Eigenschaft **TTLSEnvironmentAdvancedParmsRef** zugeordnet.

   Mit dieser Anweisung können Sie angeben, welche SSL-und TLS-Protokolle aktiviert werden sollen. Bei IBM MQ sollten Sie nur das einzige Protokoll aktivieren, das dem Cipher-Suite-Namen entspricht, der in der Anweisung TTLSCipherParms verwendet wird.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

Die vollständige Gruppe von Anweisungen lautet wie folgt und sollte auf den Richtlinienagenten angewendet werden:

```
TTLSRule                      CSQ1-TO-REMOTE
{
  LocalAddr                   ALL
  RemoteAddr                  123.456.78.9
  RemotePortRange             1414
  Jobname                     CSQ1CHIN
  Direction                   OUTBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   CLIENT
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Schritt 3: SSLCIPH aus dem z/OS -Kanal entfernen

Entfernen Sie die CipherSpec mit dem folgenden Befehl aus dem z/OS -Kanal:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Schritt 4: Kanal starten

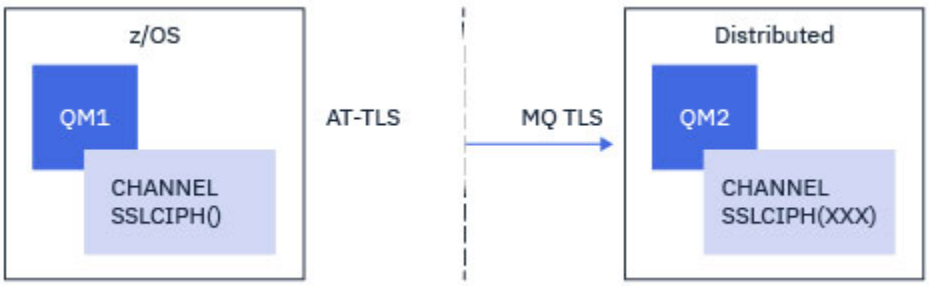
Sobald der Kanal gestartet ist, wird er eine Kombination aus AT-TLS und IBM MQ TLS verwenden.

Achtung: Die vorherigen AT-TLS-Anweisungen sind nur eine Minimalkonfiguration. Es gibt andere AT-TLS-Richtlinienanweisungen mit AT-TLS, die hier nicht dokumentiert sind, und kann je nach Bedarf mit IBM MQ verwendet werden. IBM MQ wurde jedoch nur mit den beschriebenen Richtlinien getestet.

Konfigurieren von AT-TLS auf einem abgehenden Kanal zu einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung von Alias-CipherSpecs

Wie Sie AT-TLS auf einem abgehenden Kanal von einem IBM MQ for z/OS-Queue Manager auf einen IBM MQ for Multiplatforms-Queue Manager einrichten. In diesem Fall ist der Kanal auf dem z/OS-Queue Manager ein Senderkanal, der nicht über das Attribut SSLCIPH verfügt, und der Kanal auf dem Queue Manager ohne z/OS ist ein Empfängerkanal mit dem Attribut SSLCIPH, das auf eine Alias-CipherSpec gesetzt ist.

In diesem Beispiel wird ein vorhandenes Sender-Empfänger-Kanalpaar, das die ANY_TLS13-Alias-CipherSpec verwendet, so angepasst, dass der Senderkanal AT-TLS anstelle von IBM MQ TLS verwendet.



Andere TLS-Protokolle und CiperSpecs können verwendet werden, indem kleinere Anpassungen an der Konfiguration vorgenommen werden. Andere Nachrichtenkanaltypen können, abgesehen von Clustersender- und Clusterempfängerkanälen, ohne Änderung der AT-TLS-Konfiguration verwendet werden.

Achtung: TLS 1.3 kann nur in z/OS Version 2.4 oder höher verwendet werden.

Verfahren

Schritt 1: Kanal stoppen

Schritt 2: Eine AT-TLS-Richtlinie erstellen und anwenden

Für dieses Szenario müssen Sie die folgenden AT-TLS-Anweisungen erstellen:

1. Eine Anweisung TTLSRule, mit der abgehende Verbindungen vom Kanalinitiatoradressbereich an die IP-Adresse und die Portnummer des Zielempfängerkanals abgeglichen werden sollen. Diese Werte sollten mit den Informationen übereinstimmen, die in CONNAME des Senderkanals verwendet werden. Hier wurde eine weitere Filterung eingeschlossen, die einem bestimmten Kanalinitiatorjobnamen entspricht.

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

   Die vorherige Regel stimmt mit Verbindungen überein, die mit der IP-Adresse 123.456.78.9 an Port 1414 aus dem Job CSQ1CHIN gehen.

   Weitere erweiterte Filteroptionen werden unter TTLSRule beschrieben.

2. Eine Anweisung TTLSGroupAction, mit der die Regel aktiviert wird. Der `TTLSRule` verweist auf die `TTLSGroupAction` mit der Eigenschaft **`TTLSGroupActionRef`**.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Eine TTLSEnvironmentAction-Anweisung, die der `TTLSRule` durch die Eigenschaft **`TTLSEnvironmentActionRef`** zugeordnet ist. Ein `TTLSEnvironmentAction` konfiguriert die TLS-Umgebung und gibt an, welcher Schlüsselring verwendet werden soll.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. Eine TTLSKeyringParms-Anweisung, die der TTLSEnvironmentAction durch die Eigenschaft **TTLSKeyringParmsRef** zugeordnet ist und den Schlüsselring definiert, der von AT-TLS verwendet wird.

   Der Schlüsselring sollte Zertifikate enthalten, die von dem remote Nicht-z/OS-Queue Manager anerkannt werden. Dieser Schlüsselring kann auf die gleiche Weise wie ein Schlüsselring definiert werden, der vom Kanalinitiator verwendet wird (siehe „z/OS-System für die Verwendung von TLS konfigurieren” auf Seite 274).

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Eine TTLSCipherParms-Anweisung, die der TTLSEnvironmentAction durch die Eigenschaft **TTLSCipherParmsRef** zugeordnet ist.

   Diese Anweisung muss einen oder mehrere Cipher-Suite-Namen enthalten, von denen mindestens einer mit dem Satz von CipherSpecs kompatibel sein sollte, der durch die auf dem Zielempfängerkanal verwendete Alias-CipherSpec impliziert ist.

   Anmerkung: AT-TLS-Cipher-Suite-Namen stimmen nicht unbedingt mit IBM MQ-CipherSpec-Namen überein. Es ist jedoch möglich, den AT-TLS-Cipher-Suite-Namen zu finden, der mit einem IBM MQ-CipherSpec-Namen übereinstimmt, indem der IBM MQ-CipherSpec-Name in der folgenden Tabelle gesucht wird und die hexadezimale Codespalte mit der erweiterten Zeichenspalte aus Tabelle 2 im TTLSCipherParms-Abschnitt referenziert wird.

Tabelle 81. CipherSpecs auf z/OS aus IBM MQ for z/OS 9.2.0

CipherSpec	Protokoll	Hexadezimalcode	Standardmäßig aktiviert
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Ja
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Ja
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Ja
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Ja
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Ja
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Ja
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Ja

Tabelle 81. CipherSpecs auf z/OS aus IBM MQ for z/OS 9.2.0 (Forts.)			
CipherSpec	**Protokoll**	**Hexadezimalcode**	**Standardmäßig aktiviert**
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Ja
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Ja
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Ja
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Ja
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Ja
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Ja
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Nein
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Nein
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Nein
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Nein
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Nein
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Nein
TRIPLE_DES_SHA_US	SSL v3	000A	Nein
RC4_SHA_US	SSL v3	0005	Nein
RC4_MD5_US	SSL v3	0004	Nein
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Nein
RC2_MD5_EXPORT	SSL v3	0006	Nein
NULL_SHA	SSL v3	0002	Nein
NULL_MD5	SSL v3	0001	Nein

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

Achtung: Wenn sowohl der Warteschlangenmanager als auch die AT-TLS-Richtlinie TLS 1.3unterstützen, ermöglichen nur Alias- CipherSpecs , die mindestens eine TLS 1.3 CipherSpec enthalten, das Starten des Kanals. Beispiel: Die Verwendung von ANY_TLS12 führt dazu, dass der Kanal nicht gestartet werden kann, auch wenn TTLSCipherParms TLS 1.2 CipherSpecsenthält, die Verwendung von ANY_TLS12_OR_HIGHER oder ANY_TLS13 jedoch den Start des Kanals ermöglicht. Eine Erläuterung finden Sie in „Beziehung zwischen Einstellungen für Alias-CipherSpecs” auf Seite 470.

6. Eine Anweisung TTLSEnvironmentAdvancedParms wird der `TTLSEnvironmentAction` durch die Eigenschaft **`TTLSEnvironmentAdvancedParmsRef`** zugeordnet.

   Diese Anweisung kann verwendet werden, um anzugeben, welche SSL-und TLS-Protokolle aktiviert sind, und sollte mit den Cipher-Suites in der Anweisung `TTLSCipherParms` konsistent sein.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

Die vollständige Gruppe von Anweisungen lautet wie folgt und sollte auf den Richtlinienagenten angewendet werden:

```
TTLSRule                      CSQ1-TO-REMOTE
{
  LocalAddr                   ALL
  RemoteAddr                  123.456.78.9
  RemotePortRange             1414
  Jobname                     CSQ1CHIN
  Direction                   OUTBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   CLIENT
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Schritt 3: SSLCIPH aus dem z/OS -Kanal entfernen

Entfernen Sie die CipherSpec mit dem folgenden Befehl aus dem z/OS -Kanal:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Schritt 4: Kanal starten

Sobald der Kanal gestartet ist, wird er eine Kombination aus AT-TLS und IBM MQ TLS verwenden.

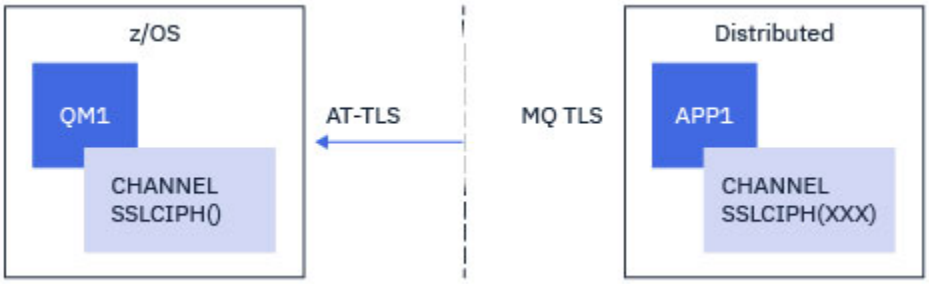
Achtung: Die vorherigen AT-TLS-Anweisungen sind nur eine Minimalkonfiguration. Es gibt andere AT-TLS-Richtlinienanweisungen mit AT-TLS, die hier nicht dokumentiert sind, und kann je nach Bedarf mit IBM MQ verwendet werden. IBM MQ wurde jedoch nur mit den beschriebenen Richtlinien getestet.

Konfigurieren von AT-TLS in einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer einzigen, benannten CipherSpec

Wie Sie AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager auf einen IBM MQ for z/OS-Queue Manager einrichten. In diesem Fall ist der Kanal auf dem z/OS-Queue Manager ein Empfängerkanal, der nicht über das Attribut SSLCIPH verfügt, und der Kanal auf dem Nicht-z/OS-Queue Manager ist ein Senderkanal, dessen Attribut SSLCIPH auf eine einzige benannte CipherSpec gesetzt ist.

Ein Beispiel für ein Beispiel unter Verwendung einer AliasCipherSpec finden Sie in „Konfigurieren von AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer Alias-CipherSpec” auf Seite 493.

In diesem Beispiel wird ein vorhandenes Sender-Empfänger-Kanalpaar, das die TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec verwendet, so angepasst, dass der Empfängerkanal AT-TLS anstelle von IBM MQ TLS verwendet.



Andere TLS-Protokolle und CipherSpecs können verwendet werden, indem kleinere Anpassungen an der Konfiguration vorgenommen werden. Andere Nachrichtenkanaltypen können, abgesehen von Clustersender- und Clusterempfängerkanälen, ohne Änderung der AT-TLS-Konfiguration verwendet werden.

Achtung: TLS 1.3 kann nur in z/OS Version 2.4 oder höher verwendet werden.

Verfahren

Schritt 1: Kanal stoppen

Schritt 2: Eine AT-TLS-Richtlinie erstellen und anwenden

Für dieses Szenario müssen Sie die folgenden AT-TLS-Anweisungen erstellen:

1. Eine Anweisung TTLSRule, mit der eingehende Verbindungen mit dem Kanalinitiatoradressbereich von der IP-Adresse des Senderkanals abgeglichen werden. Hier wurde eine weitere Filterung eingeschlossen, die einem bestimmten Kanalinitiatorjobnamen entspricht.

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

   Die vorhergehende Regel stimmt mit Verbindungen überein, die in den CSQ1CHIN-Job auf dem lokalen Port 1414 von der remote IP-Adresse 123.456.78.9 kommen.

   Weitere erweiterte Filteroptionen werden unter TTLSRule beschrieben.

2. Eine Anweisung TTLSGroupAction, mit der die Regel aktiviert wird. Der `TTLSRule` verweist auf die `TTLSGroupAction` mit der Eigenschaft **`TTLSGroupActionRef`**.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Eine TTLSEnvironmentAction-Anweisung wird dem TTLSRule durch die Eigenschaft **TTLSEnvironmentActionRef** zugeordnet. Ein TTLSEnvironmentAction konfiguriert die TLS-Umgebung und gibt an, welcher Schlüsselring verwendet werden soll.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

   AT-TLS bietet die Möglichkeit, die gegenseitige Authentifizierung zu ermöglichen, die der Verwendung des Kanalattributs SSLCAUTH entspricht. Dies geschieht, indem eine Anweisung TTLSEnvironmentAction mit dem **HandshakeRole**-Wert *ServerWithClientAuth* für die eingehende TTLSEnvironmentAction-Anweisung verwendet wird.

4. Eine TTLSKeyringParms-Anweisung wird der TTLSEnvironmentAction durch die Eigenschaft **TTLSKeyringParmsRef** zugeordnet und definiert den Schlüsselring, der von AT-TLS verwendet wird.

   Der Schlüsselring sollte Zertifikate enthalten, die von dem remote Nicht-z/OS-Queue Manager anerkannt werden. Dieser Schlüsselring kann auf die gleiche Weise wie ein Schlüsselring definiert werden, der vom Kanalinitiator verwendet wird (siehe „z/OS-System für die Verwendung von TLS konfigurieren” auf Seite 274).

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Eine TTLSCipherParms-Anweisung, die der TTLSEnvironmentAction durch die Eigenschaft **TTLSCipherParmsRef** zugeordnet ist.

   Diese Anweisung muss einen einzelnen Cipher-Suite-Namen enthalten, der dem Namen des IBM MQ CipherSpec-Namens entsprechen muss, der auf dem remote Senderkanal verwendet wird.

   Anmerkung: AT-TLS-Cipher-Suite-Namen stimmen nicht unbedingt mit IBM MQ-CipherSpec-Namen überein. Es ist jedoch möglich, den AT-TLS-Cipher-Suite-Namen zu finden, der mit einem IBM MQ-CipherSpec-Namen übereinstimmt, indem der IBM MQ-CipherSpec-Name in der folgenden Tabelle gesucht wird und die hexadezimale Codespalte mit der erweiterten Zeichenspalte aus Tabelle 2 in der TTLSCipherParms-Anweisungsgruppe referenziert wird.

Tabelle 82. CipherSpecs auf z/OS aus IBM MQ for z/OS 9.2.0

CipherSpec	Protokoll	Hexadezimalcode	Standardmäßig aktiviert
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Ja
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Ja
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Ja
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Ja
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Ja

Tabelle 82. CipherSpecs auf z/OS aus IBM MQ for z/OS 9.2.0 (Forts.)

CipherSpec	Protokoll	Hexadezimalcode	Standardmäßig aktiviert
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Ja
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Ja
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Ja
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Ja
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Ja
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Ja
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Ja
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Ja
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Nein
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Nein
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Nein
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Nein
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Nein
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Nein
TRIPLE_DES_SHA_US	SSL v3	000A	Nein
RC4_SHA_US	SSL v3	0005	Nein
RC4_MD5_US	SSL v3	0004	Nein
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Nein
RC2_MD5_EXPORT	SSL v3	0006	Nein
NULL_SHA	SSL v3	0002	Nein
NULL_MD5	SSL v3	0001	Nein

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. Eine Anweisung TTLSEnvironmentAdvancedParms wird der TTLSEnvironmentAction durch die Eigenschaft **TTLSEnvironmentAdvancedParmsRef** zugeordnet.

   Mit dieser Anweisung können Sie angeben, welche SSL-und TLS-Protokolle aktiviert werden sollen. Bei IBM MQ sollten Sie nur das einzige Protokoll aktivieren, das dem Cipher-Suite-Namen entspricht, der in der Anweisung TTLSCipherParms verwendet wird.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Die vollständige Gruppe von Anweisungen lautet wie folgt und sollte auf den Richtlinienagenten angewendet werden:

```
TTLSRule                      REMOTE-TO-CSQ1
{
  LocalAddr                   ALL
  LocalPortRange              1414
  RemoteAddr                  123.456.78.9
  Jobname                     CSQ1CHIN
  Direction                   INBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   SERVER
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Schritt 3: SSLCIPH aus dem z/OS -Kanal entfernen

Entfernen Sie die CipherSpec mit dem folgenden Befehl aus dem z/OS -Kanal:

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

Schritt 4: Kanal starten

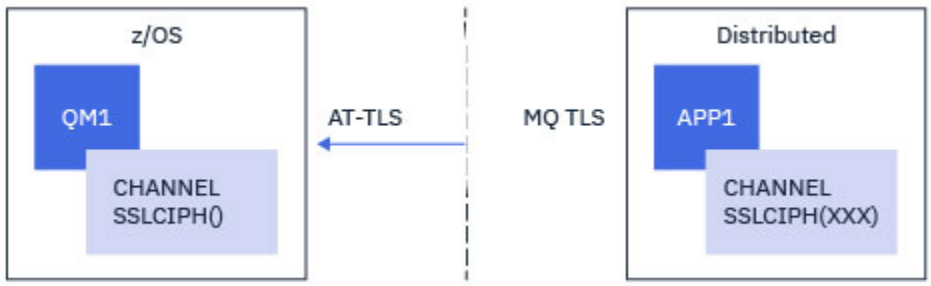
Sobald der Kanal gestartet ist, wird er eine Kombination aus AT-TLS und IBM MQ TLS verwenden.

Achtung: Die vorherigen AT-TLS-Anweisungen sind nur eine Minimalkonfiguration. Es gibt andere AT-TLS-Richtlinienanweisungen mit AT-TLS, die hier nicht dokumentiert sind, und kann je nach Bedarf mit IBM MQ verwendet werden. IBM MQ wurde jedoch nur mit den beschriebenen Richtlinien getestet.

Konfigurieren von AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager unter Verwendung einer Alias-CipherSpec

Wie Sie AT-TLS auf einem eingehenden Kanal von einem IBM MQ for Multiplatforms-Queue Manager auf einen IBM MQ for z/OS-Queue Manager einrichten. In diesem Fall ist der Kanal auf dem z/OS-Queue Manager ein Empfängerkanal, der nicht über das Attribut SSLCIPH verfügt, und der Kanal auf dem Nicht-z/OS-Queue Manager ist ein Senderkanal mit dem Attribut SSLCIPH, der auf eine Alias-CipherSpec gesetzt ist.

In diesem Beispiel wird ein vorhandenes Senderempfängerkanalpaar, das eine beliebige TLS 1.3 CipherSpec verwendet, so angepasst, dass der Empfängerkanal AT-TLS anstelle von IBM MQ TLS verwendet.



Andere TLS-Protokolle und CiperSpecs können verwendet werden, indem kleinere Anpassungen an der Konfiguration vorgenommen werden. Andere Nachrichtenkanaltypen können, abgesehen von Clustersender- und Clusterempfängerkanälen, ohne Änderung der AT-TLS-Konfiguration verwendet werden.

Achtung: TLS 1.3 kann nur in z/OS Version 2.4 oder höher verwendet werden.

Verfahren

Schritt 1: Kanal stoppen

Schritt 2: Eine AT-TLS-Richtlinie erstellen und anwenden

Für dieses Szenario müssen Sie die folgenden AT-TLS-Anweisungen erstellen:

1. Eine Anweisung TTLSRule, mit der eingehende Verbindungen mit dem Kanalinitiatoradressbereich von der IP-Adresse des Senderkanals abgeglichen werden. Hier wurde eine weitere Filterung eingeschlossen, die einem bestimmten Kanalinitiatorjobnamen entspricht.

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

Die vorhergehende Regel stimmt mit Verbindungen überein, die in den CSQ1CHIN-Job auf dem lokalen Port 1414 von der remote IP-Adresse 123.456.78.9 kommen.

Weitere erweiterte Filteroptionen werden unter TTLSRule beschrieben.

2. Eine Anweisung TTLSGroupAction, mit der die Regel aktiviert wird. Der TTLSRule verweist auf die TTLSGroupAction mit der Eigenschaft **TTLSGroupActionRef**.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Eine TTLSEnvironmentAction-Anweisung wird dem TTLSRule durch die Eigenschaft **TTLSEnvironmentActionRef** zugeordnet. Ein TTLSEnvironmentAction konfiguriert die TLS-Umgebung und gibt an, welcher Schlüsselring verwendet werden soll.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS bietet die Möglichkeit, die gegenseitige Authentifizierung zu ermöglichen, die der Verwendung des Kanalattributs SSLCAUTH entspricht. Dies geschieht, indem eine Anweisung TTLSEnvironmentAction mit dem **HandshakeRole**-Wert *ServerWithClientAuth* für die eingehende TTLSEnvironmentAction-Anweisung verwendet wird.

4. Eine TTLSKeyringParms-Anweisung wird der TTLSEnvironmentAction durch die Eigenschaft **TTLSKeyringParmsRef** zugeordnet und definiert den Schlüsselring, der von AT-TLS verwendet wird.

   Der Schlüsselring sollte Zertifikate enthalten, die von dem remote Nicht-z/OS-Queue Manager anerkannt werden. Dieser Schlüsselring kann auf die gleiche Weise wie ein Schlüsselring definiert werden, der vom Kanalinitiator verwendet wird (siehe „z/OS-System für die Verwendung von TLS konfigurieren” auf Seite 274).

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Eine TTLSCipherParms-Anweisung, die der TTLSEnvironmentAction durch die Eigenschaft **TTLSCipherParmsRef** zugeordnet ist.

   Diese Anweisung muss mindestens einen Cipher-Suite-Namen enthalten, der in der Alias-CipherSpec enthalten ist, die auf dem remote Senderkanal festgelegt ist.

   Anmerkung: AT-TLS-Cipher-Suite-Namen stimmen nicht unbedingt mit IBM MQ-CipherSpec-Namen überein. Es ist jedoch möglich, den AT-TLS-Cipher-Suite-Namen zu finden, der mit einem IBM MQ-CipherSpec-Namen übereinstimmt, indem der IBM MQ-CipherSpec-Name in der folgenden Tabelle

gesucht wird und die hexadezimale Codespalte mit der erweiterten Zeichenspalte aus Tabelle 2 in der TTLSCipherParms-Anweisungsgruppe referenziert wird.

Tabelle 83. CipherSpecs auf z/OS aus IBM MQ for z/OS 9.2.0

CipherSpec	Protokoll	Hexadezimalcode	Standardmäßig aktiviert
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Ja
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Ja
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Ja
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Ja
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Ja
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Ja
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Ja
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Ja
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Ja
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Ja
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Ja
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Ja
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Ja
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Nein
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Nein
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Nein
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Nein

Tabelle 83. CipherSpecs auf z/OS aus IBM MQ for z/OS 9.2.0 (Forts.)			
CipherSpec	**Protokoll**	**Hexadezimalcode**	**Standardmäßig aktiviert**
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Nein
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Nein
TRIPLE_DES_SHA_US	SSL v3	000A	Nein
RC4_SHA_US	SSL v3	0005	Nein
RC4_MD5_US	SSL v3	0004	Nein
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Nein
RC2_MD5_EXPORT	SSL v3	0006	Nein
NULL_SHA	SSL v3	0002	Nein
NULL_MD5	SSL v3	0001	Nein

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

Achtung: Wenn sowohl der Warteschlangenmanager als auch die AT-TLS-Richtlinie TLS 1.3unterstützen, ermöglichen nur Alias- CipherSpecs , die mindestens eine TLS 1.3 CipherSpec enthalten, das Starten des Kanals. Beispiel: Die Verwendung von ANY_TLS12 führt dazu, dass der Kanal nicht gestartet werden kann, auch wenn TTLSCipherParms TLS 1.2 CipherSpecsenthält, die Verwendung von ANY_TLS12_OR_HIGHER oder ANY_TLS13 jedoch den Start des Kanals ermöglicht. Eine Erläuterung finden Sie in „Beziehung zwischen Einstellungen für Alias-CipherSpecs” auf Seite 470.

6. Eine Anweisung TTLSEnvironmentAdvancedParms wird der TTLS`EnvironmentAction` durch die Eigenschaft **`TTLSEnvironmentAdvancedParmsRef`** zugeordnet.

   Diese Anweisung kann verwendet werden, um anzugeben, welche SSL-und TLS-Protokolle aktiviert sind, und sollte mit den Cipher-Suites in der Anweisung TTLS`CipherParms` konsistent sein.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

Die vollständige Gruppe von Anweisungen lautet wie folgt und sollte auf den Richtlinienagenten angewendet werden:

```
TTLSRule                      REMOTE-TO-CSQ1
{
  LocalAddr                   ALL
  LocalPortRange              1414
  RemoteAddr                  123.456.78.9
  Jobname                     CSQ1CHIN
  Direction                   INBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   SERVER
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Schritt 3: SSLCIPH aus dem z/OS -Kanal entfernen

Entfernen Sie die CipherSpec mit dem folgenden Befehl aus dem z/OS -Kanal:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Schritt 4: Kanal starten

Sobald der Kanal gestartet ist, wird er eine Kombination aus AT-TLS und IBM MQ TLS verwenden.

Achtung: Die vorherigen AT-TLS-Anweisungen sind nur eine Minimalkonfiguration. Es gibt andere AT-TLS-Richtlinienanweisungen mit AT-TLS, die hier nicht dokumentiert sind, und kann je nach Bedarf mit IBM MQ verwendet werden. IBM MQ wurde jedoch nur mit den beschriebenen Richtlinien getestet.

Zurücksetzen von geheimen SSL-und TLS-Schlüsseln

IBM MQ unterstützt das Zurücksetzen von geheimen Schlüsseln für Warteschlangenmanager und Clients..

Geheime Schlüssel werden zurückgesetzt, wenn eine angegebene Anzahl von verschlüsselten Datenbytes über den Kanal übertragen wurde. Wenn Kanalüberwachungssignale aktiviert sind, wird der geheime Schlüssel zurückgesetzt, bevor Daten nach einem Kanalüberwachungssignal gesendet oder empfangen werden.

Der Rücksetzwert für den Schlüssel wird immer auf der Initiierungsseite des IBM MQ-Kanals festgelegt.

Warteschlangenmanager

Verwenden Sie für einen Warteschlangenmanager den Befehl **ALTER QMGR** mit dem Parameter **SSLRKEYC** , um die Werte festzulegen, die während der Neuvereinbarung von Schlüsseln verwendet werden.

Verwenden Sie unter IBM i den Befehl **CHGMQM** mit dem Parameter **SSLRSTCNT**.

MQI-Client

Standardmäßig werden die geheimen Schlüssel von MQI-Clients nicht neu vereinbart. Sie können einen MQI-Client den Schlüssel auf drei Arten neu aushandlen. In der folgenden Liste werden die Methoden in der Reihenfolge der Priorität angezeigt. Wenn Sie mehrere Werte angeben, wird der höchste Prioritätswert verwendet.

1. Durch Verwendung des Felds KeyResetCount in der MQSCO-Struktur in einem MQCONNX-Aufruf
2. Durch Verwendung der Umgebungsvariablen MQSSLRESET
3. Durch Festlegen des Attributs "SSLKeyResetCount" in der MQI-Clientkonfigurationsdatei

Diese Variablen können auf eine ganze Zahl im Bereich von 0 bis 999 999 999 gesetzt werden, die die Anzahl der nicht verschlüsselten Byte angibt, die in einem TLS-Dialog gesendet und empfangen werden, bevor der geheime TLS-Schlüssel neu verhandelt wird. Wenn Sie den Wert 0 angeben, werden die geheimen TLS-Schlüssel nicht neu vereinbart. Wenn Sie für die Anzahl der Rücksetzungen von geheimen TLS-Schlüsseln einen Wert im Bereich von 1 Byte bis 32 KB setzen, verwenden die TLS-Kanäle als Zählerstand für die Rücksetzung des geheimen Schlüssels 32 KB. Dadurch werden überhöhte Schlüsselübersetzungen vermieden, die bei kleinen TLS-Rücksetzwerten für geheime Schlüssel auftreten würden.

Wenn ein Wert größer als null angegeben wird und Kanalüberwachungssignale für den Kanal aktiviert sind, wird auch der geheime Schlüssel neu verhandelt, bevor die Nachrichtendaten nach einem Kanalüberwachungssignal gesendet oder empfangen werden.

Die Anzahl der Byte bis zur nächsten Neuvereinbarung des geheimen Schlüssels wird nach jeder erfolgreichen Neuvereinbarung zurückgesetzt.

Vollausführliche Informationen zur MQSCO-Struktur finden Sie unter KeyResetCount (MQLONG) . Vollausführliche Informationen zu MQSSLRESET finden Sie in MQSSLRESET . Weitere Informationen zur Verwendung von TLS in der Clientkonfigurationsdatei finden Sie in der SSL-Zeilengruppe der Clientkonfigurationsdatei .

Java

Bei IBM MQ classes for Java kann eine Anwendung den geheimen Schlüssel auf eine der folgenden Arten zurücksetzen:

- Wenn Sie das Feld "sslResetCount" in der Klasse "MQEnvironment" festlegen.
- Durch Festlegen der Umgebungseigenschaft MQC.SSL_RESET_COUNT_PROPERTY in einem Hashtabellenobjekt. Anschließend weist die Anwendung die Hashtabelle dem Feld `properties` in der MQEnvironment-Klasse zu oder übergibt die Hashtabelle an ein MQQueueManager-Objekt über den zugehörigen Konstruktor.

Wenn die Anwendung mehr als eine dieser Methoden verwendet, gelten die üblichen Vorrangregeln. Informationen zu den Vorrangregeln finden Sie unter Klasse com.ibm.mq.MQEnvironment .

Der Wert des Felds 'sslResetCount' oder der Umgebungseigenschaft MQC.SSL_RESET_COUNT_PROPERTY stellt die Gesamtzahl der Bytes dar, die vom IBM MQ classes for Java-Client-Code gesendet oder empfangen wurden, bevor der gemeine Schlüssel erneut verhandelt wird. Dabei ist die Anzahl der gesendeten Bytes die Anzahl vor der Verschlüsselung und die Anzahl der empfangenen Bytes die Anzahl nach der Entschlüsselung. Die Anzahl der Byte umfasst auch Steuerinformationen, die vom IBM MQ classes for Java-Client gesendet und empfangen wurden.

Wenn der Rücksetzzähler null ist, was der Standardwert ist, wird der geheime Schlüssel nie neu vereinbart. Der Wert für die Anzahl der Rücksetzungen wird ignoriert, wenn keine Cipher-Suite angegeben wurde.

JMS

Für IBM MQ classes for JMS stellt die Eigenschaft SSLRESETCOUNT die Gesamtzahl der Byte dar, die über eine Verbindung gesendet und empfangen wurden, bevor der zur Verschlüsselung verwendete geheime Schlüssel erneut verhandelt wird. Dabei ist die Anzahl der gesendeten Bytes die Anzahl vor der Verschlüsselung und die Anzahl der empfangenen Bytes die Anzahl nach der Entschlüsselung. Die Anzahl der Bytes umfasst auch Steuerinformationen, die von IBM MQ classes for JMS gesendet und empfangen werden. Wenn Sie beispielsweise ein ConnectionFactory-Objekt konfigurieren möchten, das zum Erstellen einer Verbindung über einen TLS-fähigen MQI-Kanal mit einem geheimen Schlüssel verwendet werden kann, der nach dem Überlauf von 4 MB neu vereinbart wurde, geben Sie den folgenden Befehl an JMSAdmin aus:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

Wenn der Wert von SSLRESETCOUNT null ist (Standardwert), wird der geheime Schlüssel niemals erneut vereinbart. Wenn SSLCIPHERSUITE nicht festgelegt ist, wird die Eigenschaft SSLRESETCOUNT ignoriert.

.NET

Für nicht verwaltete .NET-Clients zeigt die ganzzahlige Eigenschaft 'SSLKeyResetCount' die Anzahl der unverschlüsselten Byte an, die in einem TLS-Dialog gesendet und empfangen werden, bevor der geheime Schlüssel neu verhandelt wird.

Weitere Informationen zur Verwendung von Objekteigenschaften in IBM MQ classes for .NET finden Sie unter Attributwerte abrufen und festlegen.

Für verwaltete .NET-Clients unterstützt die SSLStream-Klasse keine Zurücksetzung/Neuverhandlung für geheime Schlüssel. Aus Gründen der Konsistenz mit anderen IBM MQ-Clients lässt der IBM MQ-verwaltete .NET-Client die Festlegung von SSLKeyResetCount durch die Anwendungen zu. Weitere Informationen hierzu finden Sie im Abschnitt Geheimer Schlüssel zurücksetzen oder neu verhandeln .

XMS .NET

Informationen zu nicht verwalteten XMS .NET-Clients finden Sie unter Sichere Verbindungen zu einem IBM MQ-Warteschlangenmanager.

Zugehörige Verweise
ALTER QMGR
ANZEIGEN QMGR
Nachrichtenwarteschlangenmanager ändern (CHGMQM)
Nachrichten-WS-Manager anzeigen (DSPMQM)

Vertraulichkeit in Benutzerexitprogrammen implementieren

Implementieren der Vertraulichkeit in Sicherheitsexits

Sicherheitsexits können eine Rolle im Vertraulichkeitsservice spielen, indem sie den symmetrischen Schlüssel zum Verschlüsseln und Entschlüsseln der Daten, die auf dem Kanal fließen, generieren und verteilen. Eine gängige Technik hierfür verwendet die PKI-Technologie.

Ein Sicherheitsexit generiert einen Zufallsdatenwert, verschlüsselt ihn mit dem öffentlichen Schlüssel des Warteschlangenmanagers oder Benutzers, den der Sicherheitsexit für die Partnersicherheit darstellt, und sendet die verschlüsselten Daten an seinen Partner in einer Sicherheitsnachricht. Der Partner-Sicherheitsexit entschlüsselt den Zufallsdatenwert mit dem privaten Schlüssel des Warteschlangenmanagers

oder Benutzers, der bzw. der er darstellt. Jeder Sicherheitsexit kann nun den wahlfreien Datenwert verwenden, um den symmetrischen Schlüssel unabhängig von der anderen abzuleiten, indem ein Algorithmus verwendet wird, der beiden bekannt ist. Alternativ können sie den Zufallsdatenwert als Schlüssel verwenden.

Wenn der erste Sicherheitsexit seinen Partner bis zu diesem Zeitpunkt nicht authentifiziert hat, kann die nächste vom Partner gesendete Sicherheitsnachricht einen erwarteten Wert enthalten, der mit dem symmetrischen Schlüssel verschlüsselt wird. Der erste Sicherheitsexit kann nun seinen Partner authentifizieren, indem er prüft, ob der Sicherheitsexit der Partnersicherheit den erwarteten Wert korrekt verschlüsseln konnte.

Die Sicherheitsexits können diese Gelegenheit auch nutzen, um den Algorithmus für die Verschlüsselung und Entschlüsselung der Daten zu vereinbaren, die auf dem Kanal fließen, wenn mehr als ein Algorithmus für die Verwendung verfügbar ist.

Vertraulichkeit in Nachrichtenexits implementieren

Ein Nachrichtenexit auf der sendenden Seite eines Kanals kann die Anwendungsdaten in einer Nachricht verschlüsseln und ein anderer Nachrichtenexit auf der Empfangsseite des Kanals kann die Daten entschlüsseln. Aus Leistungsgründen wird normalerweise ein symmetrischer Schlüsselalgorithmus verwendet. Weitere Informationen darüber, wie der symmetrische Schlüssel generiert und verteilt werden kann, finden Sie in „Vertraulichkeit in Benutzerexitprogrammen implementieren” auf Seite 499.

Header in einer Nachricht, wie z. B. der Header der Übertragungswarteschlange, MQXQH, die den eingebetteten Nachrichtendeskriptor enthält, dürfen von einem Nachrichtenexit nicht verschlüsselt werden. Dies liegt daran, dass die Datenkonvertierung der Nachrichtenheader entweder nach dem Aufruf eines Nachrichtenexits am sendenden Ende oder vor dem Aufruf eines Nachrichtenexits am empfangenden Ende stattfindet. Wenn die Header verschlüsselt sind, schlägt die Datenkonvertierung fehl und der Kanal wird gestoppt.

Vertraulichkeit in Sende-und Empfangsexits implementieren

Sende-und Empfangsexits können verwendet werden, um die Daten, die auf einem Kanal fließen, zu verschlüsseln und zu entschlüsseln. Sie sind geeigneter als Nachrichtenexits für die Bereitstellung dieses Service aus den folgenden Gründen:

- In einem Nachrichtenkanal können Nachrichtenheader sowie die Anwendungsdaten in den Nachrichten verschlüsselt werden.
- Sende-und Empfangsexits können sowohl für MQI-Kanäle als auch für Nachrichtenkanäle verwendet werden. Parameter in MQI-Aufrufen können sensible Anwendungsdaten enthalten, die geschützt werden müssen, während sie in einem MQI-Kanal fließen. Sie können daher die gleichen Sende-und Empfangsexits für beide Arten von Kanälen verwenden.

Implementieren der Vertraulichkeit in API-Exit und API-Steuerübergabeexit

Die Anwendungsdaten in einer Nachricht können von einem API-oder API-Steuerübergabeexit verschlüsselt werden, wenn die Nachricht von der sendenden Anwendung gesendet wird und von einem zweiten Exit entschlüsselt wird, wenn die Nachricht von der empfangenden Anwendung abgerufen wird. Aus Leistungsgründen wird in der Regel ein symmetrischer Schlüsselalgorithmus für diesen Zweck verwendet. Auf der Anwendungsebene, wo viele Benutzer möglicherweise Nachrichten an die anderen senden, stellt das Problem jedoch dar, wie sichergestellt werden kann, dass nur der vorgesehene Empfänger einer Nachricht die Nachricht entschlüsseln kann. Eine Lösung ist die Verwendung eines anderen symmetrischen Schlüssels für jedes Paar von Benutzern, die Nachrichten an die anderen Benutzer senden. Diese Lösung kann jedoch schwierig und zeitaufwendig zu verwalten sein, insbesondere wenn die Benutzer zu verschiedenen Organisationen gehören. Ein Standardverfahren zur Lösung dieses Problems wird als *digitaler Kuvert* bezeichnet und verwendet die PKI-Technologie.

Wenn eine Anwendung eine Nachricht in eine Warteschlange einreiht, generiert ein API-oder API-Steuerübergabeexit einen zufälligen symmetrischen Schlüssel und verwendet den Schlüssel zum Verschlüsseln der Anwendungsdaten in der Nachricht. Der Exit verschlüsselt den symmetrischen Schlüssel mit dem

öffentlichen Schlüssel des beabsichtigten Empfängers. Sie ersetzt dann die Anwendungsdaten in der Nachricht durch die verschlüsselten Anwendungsdaten und den verschlüsselten symmetrischen Schlüssel. Auf diese Weise kann nur der vorgesehene Empfänger den symmetrischen Schlüssel und damit die Anwendungsdaten entschlüsseln. Wenn eine verschlüsselte Nachricht mehr als einen möglichen Empfänger enthält, kann der Exit eine Kopie des symmetrischen Schlüssels für jeden beabsichtigten Empfänger verschlüsseln.

Wenn verschiedene Algorithmen zum Verschlüsseln und Entschlüsseln der Anwendungsdaten für die Verwendung verfügbar sind, kann der Exit den Namen des verwendeten Algorithmus enthalten.

V 9.2.0 z/OS Vertraulichkeit für ruhende Daten in IBM MQ for z/OS mit der Dataset-Verschlüsselung

In IBM MQ for z/OS können Kunden- und Konfigurationsdaten permanent gespeichert werden, indem die Daten in die aktiven Protokolldateien, Archivprotokolldateien, Seitengruppen Bootstrap-Dateien (BSDS) und gemeinsam genutzten Nachrichtendateien (SMDS) geschrieben werden.

z/OS stellt eine effiziente und auf Richtlinien basierende Verschlüsselung von Datasets bereit. IBM MQ for z/OS unterstützt die z/OS-Dataset-Verschlüsselung für:

- Aktive Protokolldatengruppen; siehe Anmerkung „1” auf Seite 501
- Archivprotokolldatengruppen; siehe Anmerkung „2” auf Seite 501
- Seitengruppen; siehe Anmerkung „1” auf Seite 501
- BSDS; siehe Anmerkung „2” auf Seite 501
- CSQINP*-Datasets; siehe Anmerkung „2” auf Seite 501
- V 9.2.0 SMDS; siehe Anmerkung „1” auf Seite 501

Dadurch wird die Vertraulichkeit von ruhenden Daten in einem einzelnen z/OS-Warteschlangenmanager bereitgestellt.

Anmerkungen:

1. V 9.2.0 Von IBM MQ for z/OS 9.2.0aus wird die z/OS-Datenverschlüsselung für aktive Protokolle festgelegt. Seitengruppen und SMDS werden unterstützt.
2. Die Verschlüsselung von Datasets für Archivprotokolle, BSDS und CSQINP*-Datasets wird auf allen Versionen von IBM MQ for z/OS unterstützt.
3. IBM MQ Advanced Message Security stellt ein alternatives Verfahren zum Schutz von ruhenden Daten bereit. Außerdem schützt AMS auch Daten im Hauptspeicher und Daten, die gerade ausgeführt werden

Weitere Informationen zur Verschlüsselung von z/OS-Datensätzen finden Sie unter Erweiterungen für die z/OS-Datensatzverschlüsselung verwenden.

Die Konfiguration der Verschlüsselung von z/OS-Datasets liegt außerhalb der Steuerung von IBM MQ for z/OS. Die Verschlüsselungseinstellungen werden wirksam, wenn das Dataset erstellt wird.

Dadurch müssen alle vorhandenen Datasets erneut erstellt werden, damit eine neue Richtlinie zur Verschlüsselung von Datasets verwendet werden kann.

IBM MQ for z/OS kann mit einer Kombination aus verschlüsselten und nicht verschlüsselten Datasets ausgeführt werden, aber in einer Standardkonfiguration werden alle oder keine der verwendeten Datasets verschlüsselt.

V 9.2.0 z/OS Übersicht über die Schritte zum Verschlüsseln eines IBM MQ for z/OS-Datasets

Informationen zum Verschlüsseln eines IBM MQ for z/OS-Datasets.

Vorbereitende Schritte

Sie müssen sicherstellen, dass die Verschlüsselung des z/OS-Datasets in Ihrem Unternehmen korrekt konfiguriert ist. Beim Einrichten der Dataset-Verschlüsselung in einer Gruppe mit gemeinsamer Warteschlange müssen Sie die Verschlüsselung des z/OS-Datasets für die gemeinsame Datennutzung konfigurieren.

Anmerkung: Bei einem verschlüsselten z/OS-Dataset muss es sich um ein Dataset in einem erweiterten Format handeln.

Vorgehensweise

1. Konfigurieren Sie einen Verschlüsselungsschlüssel und `key-label` in RACF, um das Dataset zu verschlüsseln.
2. Erstellen Sie ein Profil für `key-label` in der Klasse RACF CSFKEYS.
3. Erteilen Sie Lesezugriff für die Benutzer-ID des Warteschlangenmanagers und alle andern Benutzer-ID, die auf die verschlüsselten Daten zugreifen müssen.

   Dazu können auch Benutzer-IDs gehören, mit denen das Druckdienstprogramm für das Dataset ausgeführt wird. Beispielsweise muss der Benutzer, der CSQUTIL SCOPY ausführt, die entsprechende Seitengruppe entschlüsseln können.
4. Ordnen Sie der Verschlüsselung `key-label` den Namen des Datasets hinzu.

   Verwenden Sie dazu eine SMS-Datenklasse oder ein RACF DFP-Segment für den Namen des Datasets oder das übergeordnete Qualifikationsmerkmal.

   Sie können `key-label` auch bei der Zuordnung des Dataset mit diesem verknüpfen.
5. Benennen Sie alle vorhandenen Datasets mit IDCAMS ALTER um.
6. Ordnen Sie das Dataset erneut mit den entsprechenden Attributen zu.
7. Kopieren Sie die Inhalte des umbenannten Datasets mit dem Befehl IDCAMS REPRO in das neue Dataset.

   Die Daten werden beim Kopieren in die Datasets verschlüsselt.
8. Wiederholen Sie die Schritte „4” auf Seite 502 bis „6” auf Seite 502 für alle weiteren Datasets, die verschlüsselt werden müssen.

V 9.2.0 z/OS Beispiel zur Verschlüsselung von aktiven Protokollen für Warteschlangenmanager

In den folgenden Abschnitten werden Sie durch den Prozess zur Aktivierung der Verschlüsselung von Datasets in vorhandenen aktiven Protokollen geführt.

Anmerkung: Der Prozess für andere Datasets entspricht weitgehend dem Prozess für aktive Protokolle.

In diesem Beispiel gilt Folgendes:

- Der Warteschlangenmanager CSQ1 wird für den Benutzer QMCSQ1 ausgeführt und verfügt über die aktiven Protokolldateien CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, usw.
- In der Hardware- und Softwareumgebung kann die Verschlüsselung von z/OS-Datasets verwendet werden
- RACF wird als SAF verwendet
- Der Warteschlangenmanager wurde gestoppt

Führen Sie den Vorgang in folgender Reihenfolge aus:

1. „Dataset-Verschlüsselungsschlüssel für den Warteschlangenmanager konfigurieren” auf Seite 503
2. „Dataset-Verschlüsselung für Protokolldatengruppen konfigurieren” auf Seite 503

V 9.2.0 z/OS Dataset-Verschlüsselungsschlüssel für den Warteschlangenmanager konfigurieren

Hier finden Sie Informationen zur Konfiguration eines Dataset-Verschlüsselungsschlüssels für einen Warteschlangenmanager.

Informationen zu diesem Vorgang

Diese Task ist eine Voraussetzung für „Dataset-Verschlüsselung für Protokolldatengruppen konfigurieren" auf Seite 503.

Vorgehensweise

1. Konfigurieren Sie einen AES-256-Bitverschlüsselungsdatenschlüssel mit einem Kennsatz, z. B. CSQ1DSKY, mit dem z/OS Key Generator Utility Program (KGUP).
2. Definieren Sie das Profil RACF CSFKEYS für den Verschlüsselungsschlüssel CSQ1DSKY, indem Sie den folgenden Befehl ausgeben:

   ```
   RDEFINE CSFKEYS CSQ1DSKY UACC(NONE)
   ```

3. Konfigurieren Sie das ICSF-Segment des Profils, damit der Schlüssel als geschützter Schlüssel verwendet werden kann, indem Sie den folgenden Befehl ausgeben:

   ```
   RALTER CSFKEYS CSQ1DSKY ICSF(SYMCPACFWRAP(YES) SYMCPACFRET(YES))
   ```

4. Ermöglichen Sie dem Warteschlangenmanager die Verwendung des Verschlüsselungsschlüssels, indem Sie ihm den Lesezugriff QMCSQ1 READ für das Profil mit folgendem Befehl erteilen:

   ```
   PERMIT CSQ1DSKY CLASS(CSFKEYS) ID(QMCSQ1) ACCESS(READ)
   ```

   Erteilen Sie die gleichen Zugriffsberechtigungen jedem Benutzer mit Verwaltungsaufgaben, der das verschlüsselte Dataset lesen oder schreiben muss.
5. Aktualisieren Sie die Klasse CSFKEYS mit folgendem Befehl:

   ```
   SETROPTS RACLIST(CSFKEYS) REFRESH
   ```

Nächste Schritte

Konfigurieren Sie die Dataset-Verschlüsselung für die Datasets wie unter „Dataset-Verschlüsselung für Protokolldatengruppen konfigurieren" auf Seite 503 beschrieben.

V 9.2.0 z/OS Dataset-Verschlüsselung für Protokolldatengruppen konfigurieren

Hier finden Sie Informationen zum Konfigurieren der Verschlüsselung in den Protokolldatengruppen.

Vorbereitende Schritte

Stellen Sie sicher, dass Sie die Informationen in

Übersicht über die Schritte zur Verschlüsselung eines IBM MQ for z/OS-Datasets gelesen und die folgende Prozedur ausgeführt haben:
„Dataset-Verschlüsselungsschlüssel für den Warteschlangenmanager konfigurieren" auf Seite 503

Informationen zu diesem Vorgang

Bei dieser Methode wird das DFP-Segment eines generischen RACF-Profils verwendet, damit Sie den Verschlüsselungsschlüssel für alle neuen Datasets verwenden können, die mit dem Profil übereinstimmen.

Alternativ können Sie eine SMS-Datenklasse konfigurieren und verwenden oder den Schlüsselkennsatz direkt bei der Zuordnung des Datasets angeben.

Wie zuvor beschrieben wird der Warteschlangenmanager CSQ1 in diesem Beispiel für den Benutzer QMCSQ1 ausgeführt und verfügt über die aktiven Protokolldateien CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, usw.

Vorgehensweise

1. Falls das generische Profil nicht vorhanden ist, erstellen Sie es mit folgendem Befehl:

   ```
   ADDSD 'CSQ1.LOGS.*' UACC(NONE)
   ```

2. Ermöglichen Sie dem Warteschlangenmanagerbenutzer den Zugriff für Änderungen des Profils, indem Sie folgenden Befehl ausgeben:

   ```
   PERMIT 'CSQ1.LOGS.*' ID(QMCSQ1) ACCESS(ALTER)
   ```

   Erteilen Sie außerdem den entsprechenden Zugriff, der für einen Benutzer mit Verwaltungsaufgaben erforderlich ist.

3. Fügen Sie das DFP-Segment mit der Bezeichnung des Verschlüsselungsschlüssels hinzu, indem Sie den folgenden Befehl ausgeben:

   ```
   ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
   ```

   Anmerkung: Sie müssen den gleichen Verschlüsselungsschlüssel verwenden, den Sie beim Konfigurieren des Dataset-Verschlüsselungsschlüssels für den Warteschlangenmanager verwendet haben.

4. Aktualisieren Sie die generischen Dataset-Profile, indem Sie den folgenden Befehl ausgeben:

   ```
   SETROPTS GENERIC(DATASET) REFRESH
   ```

5. Benennen Sie jede Protokolldatei für eine Sicherungskopie um und stellen Sie die Daten mithilfe von IDCAMS wieder her. Mit folgendem JCL-Fragment wird CSQ1.LOGS.LOGCOPY1.DS001 umgewandelt:

   a) Benennen Sie das Dataset für eine Sicherungskopie um

   ```
   //RENAME   EXEC PGM=IDCAMS,REGION=0M
   //SYSPRINT DD SYSOUT=*
   //SYSIN    DD *
     /*-------------------------------------------------------------*/
     /* RENAME DATASET TO BACKUP                                    */
     /*-------------------------------------------------------------*/
     ALTER 'CSQ1.LOGS.LOGCOPY1.DS001'                  -
           NEWNAME('CSQ1.BAK.LOGS.LOGCOPY1.DS001')
   ```

   b) Definieren Sie das Dataset erneut.

   Das neue Dataset wird auf Grundlage des RACF-Profils verschlüsselt.

   Anmerkung: Ersetzen Sie ++EXTDCLASS++ durch den Namen der Datenklasse für das erweiterte Format, die Sie für das Dataset verwenden möchten.

   ```
   //REDEFINE EXEC PGM=IDCAMS,REGION=0M
   //SYSPRINT DD SYSOUT=*
   //SYSIN    DD *
     /*-------------------------------------------------------------*/
     /* REDEFINE THE DATASET                                        */
     /*-------------------------------------------------------------*/
     DEFINE CLUSTER                                  -
           (NAME(CSQ1.LOGS.LOGCOPY1.DS001)           -
           LINEAR                                    -
           SHAREOPTIONS(2 3)                         -
           MODEL(CSQ1.BAK.LOGS.LOGCOPY1.DS001)       -
           DATACLAS(++EXTDCLASS++))
   ```

   c) Kopieren Sie die Daten aus der Sicherungskopie in das neu erstellte Dataset.

   In diesem Schritt werden die Daten verschlüsselt:

```
//RESTORE  EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RESTORE DATA INTO ENCRYPTED LOG                             */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)      -
        OUTDATASET(CSQ1.LOGS.LOGCOPY1.DS001)
```

Nächste Schritte

Wiederholen Sie den Schritt „5” auf Seite 504 für alle aktiven Protokolldatengruppen.

Es ist nur ein einziger Verschlüsselungsschlüssel erforderlich und alle Datasets können dem gleichen Schlüsselkennsatz zugeordnet werden.

Starten Sie den Warteschlangenmanager CSQ1 erneut. Überprüfen Sie mithilfe der Ausgabe des Befehls DISPLAY LOG, dass die Protokolldatengruppen verschlüsselt wurden.

V 9.2.0 z/OS Hinweise zur Verschlüsselung von z/OS-Datasets in einer Gruppe mit gemeinsamer Warteschlange

Jeder Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange (Queue sharing group, QSG) muss die Protokolle , BSDS und gemeinsam genutzten Nachrichtendateien (Shared message data sets, SMDS) jedes anderen Warteschlangenmanagers in der QSG lesen können.

Das bedeutet, dass jedes System, in dem ein Mitglied der QSG ausgeführt werden kann, die Voraussetzungen für die Verschlüsselung von z/OS-Datasets erfüllen muss und dass alle Schlüsselkennsätze und Verschlüsselungsschlüssel, mit denen die Datasets für jeden Warteschlangenmanager in der QSG geschützt werden, auf jedem System verfügbar sein müssen.

Ein Warteschlangenmanager vor IBM MQ for z/OS 9.1.4 kann nicht auf die verschlüsselten aktiven Protokolldateien zugreifen.

Ein Warteschlangenmanager vor IBM MQ for z/OS 9.1.5 kann nicht auf verschlüsselte SMDS zugreifen.

Bevor Sie die Verschlüsselung von z/OS-Dataset verwenden, sollten Sie alle Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange auf mindestens IBM MQ for z/OS 9.1.5 migrieren.

Wenn ein Warteschlangenmanager in einer QSG mit einer verschlüsselten aktiven Protokolldatei gestartet wird und ein anderer Warteschlangenmanager in der QSG zwar gestartet wurde, aber zuletzt nicht mit einer Version von IBM MQ for z/OS, in der verschlüsselte aktive Protokolle unterstützt werden, wird der Warteschlangenmanager mit dem verschlüsselten aktiven Protokoll abnormal mit dem Abbruchcode 5C6-00F50033 beendet.

Sie können eine QSG so konvertieren, dass verschlüsselte aktive Protokolle und SMDS ohne vollständigen Ausfall verwendet werden können; dies geschieht folgendermaßen:

1. Migration jedes Warteschlangenmanagers jeweils auf mindestens IBM MQ for z/OS 9.1.5.
2. Konvertierung von aktiven Protokollen in verschlüsselte Datasets für jeden Warteschlangenmanager. Dazu muss der Warteschlangenmanager beendet und anschließend erneut gestartet werden.

   Gleichzeitig ist es wahrscheinlich, dass Seitengruppen und Archivprotokolle auch für verschlüsselte Datensätze aktiviert werden würden, aber dies wirkt sich nicht auf die QSG-Migration aus.

   Die Vorgehensweise zum Konvertieren jedes Datasets wird im Abschnitt „Beispiel zur Verschlüsselung von aktiven Protokollen für Warteschlangenmanager” auf Seite 502 beschrieben.
3. Konvertierung von SMDS in verschlüsselte Datasets nacheinander für jede einzelne CF-Struktur; gehen Sie dazu folgendermaßen vor:

   a. Geben Sie den Befehl RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(Strukturname) aus, um den Zugriff des Warteschlangenmanagers auf SMDS auszusetzen.

Beachten Sie, dass die Daten in den gemeinsam genutzten Warteschlangen, die SMDS zugeordnet sind, in dieser Zeit vorübergehend nicht verfügbar sind.

b. Konvertieren Sie jedes Dataset, aus denen SMDS gebildet werden, mithilfe der im Abschnitt „Beispiel zur Verschlüsselung von aktiven Protokollen für Warteschlangenmanager” auf Seite 502 beschriebenen Vorgehensweise in verschlüsselte Datasets.

c. Geben Sie den Befehl RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(Strukturname) aus, um den Zugriff des Warteschlangenmanagers auf die SMDS fortzusetzen.

Achtung: Sie sollten den Warteschlangenmanager ordnungsgemäß beenden, bevor Sie die Protokolle konvertieren, und die Wiederherstellung der Coupling-Facility-Struktur ist möglicherweise während der Konvertierung nicht möglich, da die aktiven Protokolldateien vorübergehend nicht verfügbar sind.

V 9.2.0 z/OS Hinweise zur Rückwärtsmigration bei der Verwendung der Verschlüsselung für z/OS-Datasets

Sie müssen die folgenden Aspekte bei der Rückwärtsmigration eines Warteschlangenmanagers berücksichtigen, der über ein oder mehrere verschlüsselte Datasets verfügt.

Die Verschlüsselung von z/OS-Datasets wird in den folgenden IBM MQ for z/OS-Datasets unterstützt:

- Aktive Protokolldateien
- Archivprotokolldateien
- Seitengruppen
- BSDS
- SMDS
- CSQINP*-Datasets

Es gibt keine Hinweise zur Rückwärtsmigration für BSDS, Archivprotokolle oder CSINP*-Datasets.

Es gibt allerdings Hinweise zu den folgenden Datasets:

- SMDS
- Seitengruppe
- Aktives Protokoll

Die Verwendung dieser Datasets mit der Verschlüsselung von z/OS-Datasets wird in in den Long Term Support-Releases von IBM MQ for z/OS 9.1.0 und früher nicht unterstützt.

Vor der Rückwärtsmigration müssen alle Verschlüsselungsrichtlinien für SMDS, Seitengruppen und aktive Protokolldatengruppen entfernt und die Daten müssen entschlüsselt werden. Dieser Vorgang wird im Abschnitt „Dataset-Verschlüsselung auf einem Dataset entfernen” auf Seite 506 beschrieben.

Achtung: Wenn der Warteschlangenmanager, für den eine Rückwärtsmigration durchgeführt werden soll, Teil einer Gruppe mit gemeinsamer Warteschlange (Queue sharing group, QSG) ist, lesen Sie zuerst den Abschnitt „Hinweise zur Gruppe mit gemeinsamer Warteschlange” auf Seite 508.

Dataset-Verschlüsselung auf einem Dataset entfernen

In diesem Beispiel wird beschrieben, wie die Dataset-Verschlüsselung aus der Protokolldatengruppe CSQ1.LOGS.LOGCOPY1.DS001 entfernt wird. Sie können eine entsprechende Vorgehensweise für V 9.2.0 SMDS und Seitengruppen anwenden.

In diesem Beispiel wird Folgende vorausgesetzt:

- RACF wird als SAF verwendet
- Der Warteschlangenmanager, der das Dataset verwendet, wurde gestoppt
- Die Bezeichnung des Verschlüsselungsschlüssels wurde dem generischen RACF-Profil CSQ1.LOGS.* zugeordnet.

Gehen Sie wie folgt vor:

1. Kopieren Sie die Daten aus dem Dataset in eine Sicherungsdatei.

   a. Definieren Sie eine Sicherungsdatei, die keiner Bezeichnung für einen Verschlüsselungsschlüssel zugeordnet ist.

      Anmerkung: Ersetzen Sie ++EXTDCLASS++ durch den Namen der Datenklasse für das erweiterte Format, die Sie für das Dataset verwenden möchten.

      ```
      //DEFINE EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* DEFINE UNENCRYPTED DATA SET                                 */
        /*-------------------------------------------------------------*/
        DEFINE CLUSTER                                  -
               (NAME(CSQ1.BAK.LOGS.LOGCOPY1.DS001)      -
               LINEAR                                   -
               SHAREOPTIONS(2 3)                        -
               MODEL(CSQ1.LOGS.LOGCOPY1.DS001)          -
               DATACLAS(++EXTDCLASS++))
      /*
      ```

   b. Kopieren Sie die Daten aus dem ursprünglichen Dataset in die Sicherungsdatei. In diesem Schritt werden die Daten entschlüsselt.

      ```
      //COPY     EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* COPY DATA INTO UNENCRYPTED DATA SET                         */
        /*-------------------------------------------------------------*/
        REPRO INDATASET(CSQ1.LOGS.LOGCOPY1.DS001)          -
              OUTDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)
      /*
      ```

   c. Löschen Sie das ursprüngliche Dataset.

      ```
      //DELETE   EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* DELETE ORIGINAL                                             */
        /*-------------------------------------------------------------*/
        DELETE ('CSQ1.LOGS.LOGCOPY1.DS001')
      /*
      ```

   d. Geben Sie der Sicherungsdatei den Namen des ursprünglichen Datasets. Die Daten bleiben unverschlüsselt.

      ```
      //RENAME   EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* RENAME UNENCRYPTED DATA SET                                 */
        /*-------------------------------------------------------------*/
        ALTER CSQ1.BAK.LOGS.LOGCOPY1.DS001'                -
               NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001)
        ALTER 'CSQ1.BAK.LOGS.LOGCOPY1.DS001.*'              -
               NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001.*')
      /*
      ```

2. Wiederholen Sie diesen Prozess optional für andere Dateien, denen ein Chiffrierschlüsselkennsatz über CSQ1.LOGS.* generisches Profil.
3. Optional, wenn alle Dateien dem CSQ1.LOGS.* Das generische Profil wurde entschlüsselt. Entfernen Sie das DATAKEY, das dem generischen Profil zugeordnet ist, indem Sie den folgenden Befehl absetzen:

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

4. Aktualisieren Sie die generischen Dataset-Profile, indem Sie den folgenden Befehl ausgeben:

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. Starten Sie den Warteschlangenmanager erneut.
6. Wenn der Verschlüsselungsschlüssel nicht mehr benötigt wird, löschen Sie ihn und löschen Sie das zugehörige RACF-Profil in der Klasse CSFKEYS.

Hinweise zur Gruppe mit gemeinsamer Warteschlange

Wenn für einen Warteschlangenmanager, der Teil einer Gruppe mit gemeinsamer Warteschlange ist, eine Rückwärtsmigration auf eine Version von IBM MQ for z/OS durchgeführt werden soll, in der die Verschlüsselung von Datasets nicht unterstützt wird, müssen die Verschlüsselungsrichtlinien für Datasets für alle aktiven Protokolldatengruppen und SMDS aller Warteschlangenmanager in der QSG entfernt und die zugehörigen Daten müssen entschlüsselt werden.

Dies gilt unabhängig davon, ob für einen einzelnen Member der QSG oder für alle Member eine Rückwärtsmigration durchgeführt werden soll.

Wenn Sie Verschlüsselungsrichtlinie entfernen und Daten entschlüsseln möchten, ohne die Gruppe mit gemeinsamer Warteschlange vollständig zu unterbrechen, gehen Sie folgendermaßen vor:

1. Beenden Sie jeden Warteschlangenmanager in der QSG der Reihe nach, um die Verschlüsselungsrichtlinien zu entfernen und die Daten aus den aktiven Protokollen zu entschlüsseln, wie im Abschnitt „Dataset-Verschlüsselung auf einem Dataset entfernen” auf Seite 506 beschrieben wird.

   Wenn eine Rückwärtsmigration für den Warteschlangenmanager vorgenommen werden soll, sollte auch die zugehörige Seitengruppe zu diesem Zeitpunkt entschlüsselt werden. Starten Sie anschließend den Warteschlangenmanager erneut.

2. V 9.2.0 Entfernen Sie die Verschlüsselungsrichtlinien und entschlüsseln Sie die Daten für die SMDS für jede einzelne CF-Struktur der Reihe nach, indem Sie folgendermaßen vorgehen:

   a. Geben Sie folgenden Befehl aus:

   ```
   RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name)
   ```

   Damit wird der Zugriff des Warteschlangenmanager auf die SMDS ausgesetzt. Während dieser Zeit sind die Daten in den gemeinsam genutzten Warteschlangen, die den SMDS zugeordnet sind, vorübergehend nicht verfügbar.

   b. Folgen Sie dem Vorgang im Abschnitt „Dataset-Verschlüsselung auf einem Dataset entfernen” auf Seite 506 für jedes Dataset, aus dem die SMDS gebildet werden.

   c. Geben Sie folgenden Befehl aus:

   ```
   RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name)
   ```

   Damit wird der Zugriff des Warteschlangenmanager auf die SMDS fortgesetzt.

Verschlüsselung von z/OS -Datasets mit einem Warteschlangenmanager verwenden, der die Verschlüsselung nicht unterstützt

Wenn Sie versehentlich eine Rückwärtsmigration eines Warteschlangenmanagers auf eine Version von IBM MQ for z/OS durchführen, in der die Verschlüsselung von Datasets nicht unterstützt wird, und Sie vergessen, die Verschlüsselungsrichtlinien zu entfernen und die Daten zu entschlüsseln, wird ein Fehler angezeigt, wenn der Warteschlangenmanager versucht, auf das Dataset zuzugreifen.

Der Fehler ist vom Typ des Datasets abhängig und wird in der folgende Tabelle angezeigt.

Anmerkung: Wenn einer oder mehrere dieser Fehler auftreten, müssen Sie für das beteiligte Dataset den im Abschnitt „Dataset-Verschlüsselung auf einem Dataset entfernen” auf Seite 506 beschriebenen Prozessen folgen. Diese können ausgeführt werden, ohne dass die Version von IBM MQ for z/OS geändert werden muss.

Datensatz	Fehler, wenn der Warteschlangenmanager die Verschlüsselung von z/OS-Datasets nicht unterstützt
Seitengruppe 0	Abnormale Beendigung 5C6-00C91400 beim Start des Warteschlangenmanagers
Seitengruppen 1-99	MQRC 2193 “Seitengruppenfehler” beim Zugriff auf die Seitengruppe, z. B. in MQPUT
Aktives Protokoll	Abnormale Beendigung 5C6-00E80084 beim Start des Warteschlangenmanagers
V 9.2.0 SMDS	Nachricht IEC161I-122 wird protokolliert “The data set has a KEYLABEL, but the user did not specify that the application could handle encryption”. SMDS wird gekennzeichnet mit AVAIL(ERROR).

Datenintegrität von Nachrichten

Um die Datenintegrität zu gewährleisten, können Sie verschiedene Typen von Benutzerexitprogrammen verwenden, um Nachrichtendigests oder digitale Signaturen für Ihre Nachrichten bereitzustellen.

Datenintegrität

Datenintegrität in Nachrichten implementieren

Wenn Sie TLS verwenden, bestimmt Ihre Auswahl von CipherSpec die Ebene der Datenintegrität im Unternehmen. Wenn Sie den IBM MQ Advanced Message Service (AMS) verwenden, können Sie die Integrität für eine eindeutige Nachricht angeben.

Datenintegrität in Nachrichtenexits implementieren

Eine Nachricht kann von einem Nachrichtenexit am sendenden Ende eines Kanals digital signiert werden. Die digitale Signatur kann dann von einem Nachrichtenexit auf der Empfangsseite eines Kanals überprüft werden, um festzustellen, ob die Nachricht absichtlich geändert wurde.

Ein bestimmter Schutz kann bereitgestellt werden, indem anstelle einer digitalen Signatur ein Nachrichtendigest verwendet wird. Ein Nachrichten-Digest kann gegen gelegentliche oder wahllose Manipulation von Manipulationen wirksam sein, verhindert jedoch nicht, dass die Nachrichten die Nachricht ändern oder ersetzen und einen völlig neuen Digest für sie generieren. Dies gilt insbesondere dann, wenn der Algorithmus, der zum Generieren des Nachrichten-Digest verwendet wird, ein bekannter Algorithmus ist.

Datenintegrität in Sende-und Empfangsexits implementieren

In einem Nachrichtenkanal sind Nachrichtenexits besser geeignet, diesen Service bereitzustellen, da ein Nachrichtenexit Zugriff auf eine ganze Nachricht hat. In einem MQI-Kanal können Parameter in MQI-Aufrufen Anwendungsdaten enthalten, die geschützt werden müssen, und nur Sende-und Empfangsexits können diesen Schutz bereitstellen.

Implementieren der Datenintegrität im API-Exit oder API-Steuerübergabeexit

Eine Nachricht kann von einem API-oder API-Steuerübergabeexit digital signiert werden, wenn die Nachricht von der sendenden Anwendung gestellt wird. Die digitale Signatur kann dann von einem zweiten Exit überprüft werden, wenn die Nachricht von der empfangenden Anwendung abgerufen wird, um festzustellen, ob die Nachricht absichtlich geändert wurde.

Ein bestimmter Schutz kann bereitgestellt werden, indem anstelle einer digitalen Signatur ein Nachrichtendigest verwendet wird. Ein Nachrichten-Digest kann gegen gelegentliche oder wahllose Manipulation von Manipulationen wirksam sein, verhindert jedoch nicht, dass die Nachrichten die Nach-

richt ändern oder ersetzen und einen völlig neuen Digest für sie generieren. Dies gilt insbesondere dann, wenn der Algorithmus, der zum Generieren des Nachrichten-Digest verwendet wird, ein bekannter Wert ist,

Weitere Informationen

Im Abschnitt „CipherSpecs aktivieren” auf Seite 448 finden Sie weitere Informationen zur Gewährleistung der Datenintegrität.

Zugehörige Tasks

Verbinden von zwei WS-Managern mit TLS

Client sicher mit einem WS-Manager verbinden

Prüfprotokollierungs

Sie können mithilfe von Ereignisnachrichten auf Sicherheitseinbrüche oder unbefugte Zugriffe überprüfen. Sie können die Sicherheit Ihres Systems auch mit dem IBM MQ Explorer überprüfen.

Überprüfen Sie die Ereignisnachrichten, die von Ihren Warteschlangenmanagern erstellt werden, insbesondere Berechtigungsereignisnachrichten, um Versuche zu erkennen, nicht autorisierte Aktionen auszuführen, wie z. B. die Verbindung zu einem Warteschlangenmanager oder eine Nachricht in eine Warteschlange zu stellen. Weitere Informationen zu Ereignisnachrichten des Warteschlangenmanagers finden Sie unter Warteschlangenmanagerereignisse und weitere Informationen zur Ereignisüberwachung im Allgemeinen finden Sie im Abschnitt Ereignisüberwachung .

Cluster sicher halten

Autorisieren oder Verhindern, dass WS-Manager Cluster verbinden oder Nachrichten in Clusterwarteschlangen stellen. Erzwingen Sie, dass ein WS-Manager einen Cluster verlässt. Wenn Sie TLS für Cluster konfigurieren, müssen Sie einige zusätzliche Aspekte berücksichtigen.

Unberechtigte Warteschlangenmanager stoppen, die Nachrichten senden

Verhindern Sie, dass nicht berechtigte WS-Manager Nachrichten an Ihren Warteschlangenmanager senden, indem Sie einen Kanalsicherheitsexit verwenden.

Vorbereitende Schritte

Clustering hat keine Auswirkungen auf die Art und Weise, wie Sicherheitsexits funktionieren. Sie können den Zugriff auf einen Warteschlangenmanager auf die gleiche Weise wie in einer verteilten Warteschlangenumgebung einschränken.

Informationen zu diesem Vorgang

Verhindern Sie, dass die ausgewählten Warteschlangenmanager Nachrichten an Ihren Warteschlangenmanager senden:

Vorgehensweise

1. Definieren Sie ein Kanalsicherheitsexitprogramm in der Kanaldefinition CLUSRCVR .
2. Schreiben Sie ein Programm, das Warteschlangenmanager authentifiziert, die versuchen, Nachrichten auf Ihrem Clusterempfängerkanal zu senden, und verweigert ihnen den Zugriff, wenn sie nicht berechtigt sind.

Nächste Schritte

Kanalsicherheitsexitprogramme werden beim Start und bei der Beendigung des Nachrichtenkanalagenten aufgerufen.

Stoppen von nicht berechtigten Warteschlangenmanagern, die Nachrichten in Ihre Warteschlangen stellen

Verwenden Sie das Attribut "channel put authority" auf dem Clusterempfängerkanal, um nicht berechtigte Warteschlangenmanager zu stoppen, die Nachrichten in Ihre Warteschlangen einreihen. Autorisieren Sie einen fernen Warteschlangenmanager, indem Sie die Benutzer-ID in der Nachricht mithilfe von RACF unter z/OS oder mit dem OAM auf anderen Plattformen prüfen.

Informationen zu diesem Vorgang

Verwenden Sie die Sicherheitseinrichtungen auf einer Plattform und die Zugriffssteuerungsmechanismen in IBM MQ, um den Zugriff auf Warteschlangen zu steuern.

Vorgehensweise

1. Um zu verhindern, dass bestimmte WS-Manager Nachrichten in eine Warteschlange stellen, verwenden Sie die Sicherheitsfunktionen, die auf Ihrer Plattform verfügbar sind.

   Beispiel:

   - RACF oder andere externe Sicherheitsmanager unter IBM MQ for z/OS
   - Der Objektberechtigungsmanager (OAM) auf anderen Plattformen.
2. Verwenden Sie die Berechtigung put, PUTAUT , Attribut in der Kanaldefinition CLUSRCVR .

   Mit dem Attribut PUTAUT können Sie angeben, welche Benutzer-IDs verwendet werden sollen, um die Berechtigung zum Angeben einer Nachricht in eine Warteschlange zu erstellen.

   Die Optionen für das Attribut PUTAUT sind:

   DEF
   Verwenden Sie die Standardbenutzer-ID. Unter z/OS kann in der Prüfung die vom Netz empfangene und die von MCAUSER abgeleitete Benutzer-ID verwendet werden.

   CTX
   Verwenden Sie die Benutzer-ID in den Kontextinformationen, die der Nachricht zugeordnet sind. Unter z/OS kann in der Prüfung die vom Netz empfangene Benutzer-ID und/oder die von MCAUSER abgeleitete Benutzer-ID verwendet werden. Verwenden Sie diese Option, wenn der Link vertrauenswürdig und authentifiziert ist.

   ONLYMCA (nur z/OS)
   Wie bei DEF wird die vom Netz empfangene Benutzer-ID nicht verwendet. Verwenden Sie diese Option, wenn der Link nicht vertrauenswürdig ist. Sie möchten nur eine bestimmte Gruppe von Aktionen für sie zulassen, die für den MCAUSER definiert sind.

   ALTMCA (nur z/OS)
   Wie bei CTX , aber jede aus dem Netz empfangene Benutzer-ID wird nicht verwendet.

Berechtigung zum Einreihen von Nachrichten in ferne Clusterwarteschlangen berechtigen

Richten Sie unter z/OS die Berechtigung zum Einreihen in eine Clusterwarteschlange mit RACF ein. Berechtigen Sie auf anderen Plattformen den Zugriff zum Herstellen einer Verbindung zu den Warteschlangenmanagern und zum Einlegen in die Warteschlangen auf diesen Warteschlangenmanagern.

Informationen zu diesem Vorgang

Das Standardverfahren besteht darin, eine Zugriffssteuerung für dieSYSTEM.CLUSTER.TRANSMIT.QUEUEdurchzuführen. Beachten Sie, dass dieses Verhalten auch dann gilt, wenn Sie mehrere Übertragungswarteschlangen verwenden.

Das in diesem Abschnitt beschriebene Verfahren gilt nur, wenn Sie das**ClusterQueueAccessControl** Attribut in der qm.iniDatei als *RQMName* konfiguriert haben, wie im Abschnitt Sicherheits-Stanza beschrieben, und den Warteschlangenmanager neu gestartet haben.

Prozedur

- Geben Sie für z/OS die folgenden Befehle aus:

```
RDEFINE MQQUEUE QMgrName.QUEUE. QueueName UACC(NONE)
PERMIT QMgrName.QUEUE. QueueName CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

- Geben Sie auf Systemen mit AIX, Linux, and Windows die folgenden Befehle aus:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
```

- Geben Sie für IBM i die folgenden Befehle aus:

```
GRTMQMAUT OBJ(' QMgrName ') OBJTYPE(*MQM) USER(GroupName) AUT(*CONNECT)
GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

Der Benutzer kann Nachrichten nur in die angegebene Clusterwarteschlange und in keine anderen Clusterwarteschlangen stellen.

Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers. Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

GroupName
Der Name der Gruppe, auf die der Zugriff erteilt werden soll.

QueueName
Name der Warteschlange oder des generischen Profils, für die Berechtigungen geändert werden sollen.

Nächste Schritte

Wenn Sie eine Empfangswarteschlange für Antworten angeben, wenn Sie eine Nachricht in eine Clusterwarteschlange einlegen, muss die konsumiende Anwendung berechtigt sein, die Antwort zu senden. Legen Sie diese Berechtigung fest, indem Sie die Anweisungen im Abschnitt „Berechtigung zum Einlegen von Nachrichten in eine ferne Clusterwarteschlange erteilen” auf Seite 423 befolgen.

Zugehörige Konzepte
Sicherheitszeilengruppe in 'qm.ini'

Verhindern, dass WS-Manager in einen Cluster

Wenn ein Rogue-WS-Manager einem Cluster beitritt, ist es schwierig zu verhindern, dass er Nachrichten empfängt, die er nicht empfangen soll.

Vorgehensweise

Wenn Sie sicherstellen wollen, dass nur bestimmte berechtigte WS-Manager einem Cluster beitreten, haben Sie die Wahl zwischen drei Verfahren:

- Mithilfe von Kanalauthentifizierungsdatensätzen können Sie die Clusterkanalverbindung basierend auf der fernen IP-Adresse, dem Namen des fernen Warteschlangenmanagers oder dem vom fernen System bereitgestellten TLS-DN blockieren.

- Schreiben Sie ein Exitprogramm, um zu verhindern, dass nicht berechtigte WS-Manager in SYSTEM.CLUSTER.COMMAND.QUEUE schreiben. Beschränken Sie den Zugriff auf SYSTEM.CLUSTER.COMMAND.QUEUE nicht so, dass kein Warteschlangenmanager Daten schreiben kann, oder Sie verhindern, dass ein WS-Manager dem Cluster beitreten kann.
- Ein Sicherheitsexitprogramm in der CLUSRCVR -Kanaldefinition.

Sicherheitsexits auf Clusterkanälen

Zusätzliche Aspekte bei der Verwendung von Sicherheitsexits auf Clusterkanälen.

Informationen zu diesem Vorgang

Wenn ein Clustersenderkanal zum ersten gestartet wird, verwendet er Attribute, die manuell von einem Systemadministrator definiert werden. Wenn der Kanal gestoppt und erneut gestartet wird, nimmt er die Attribute aus der entsprechenden Kanaldefinition des Clusterempfängers auf. Die ursprüngliche Clustersenderkanaldefinition wird mit den neuen Attributen überschrieben, einschließlich des Attributs SecurityExit .

Vorgehensweise

1. Sie müssen einen Sicherheitsexit sowohl auf der Clustersenderseite als auch auf dem Clusterempfängerende eines Kanals definieren.

   Die erste Verbindung muss mit einem Handshake für den Sicherheitsexit hergestellt werden, auch wenn der Name des Sicherheitsexits von der Clusterempfängerdefinition gesendet wird.
2. Überprüfen Sie den PartnerName in der MQCXP -Struktur im Sicherheitsexit.

   Der Exit muss zulassen, dass der Kanal nur gestartet wird, wenn der Partnerwarteschlangenmanager berechtigt ist.
3. Entwerfen Sie den Sicherheitsexit auf der Clusterempfängerdefinition, der vom Empfänger eingeleitet werden soll.
4. Wenn Sie ihn als Absender entwerfen, kann ein nicht berechtigter Warteschlangenmanager ohne Sicherheitsexit dem Cluster beitreten, da keine Sicherheitsprüfungen ausgeführt werden.

   Erst wenn der Kanal gestoppt und erneut gestartet wird, kann der Name SCYEXIT von der Cluster-Empfänger-Definition und den vollständigen Sicherheitsprüfungen gesendet werden.
5. Verwenden Sie den folgenden Befehl, um die momentan im Gebrauch angegebene Clustersenderkanaldefinition anzuzeigen:

   ```
   DISPLAY CLUSQMGR( queue manager ) ALL
   ```

   Mit dem Befehl werden die Attribute angezeigt, die über die Clusterempfängerdefinition gesendet wurden.
6. Verwenden Sie den folgenden Befehl, um die ursprüngliche Definition anzuzeigen:

   ```
   DISPLAY CHANNEL( channel name ) ALL
   ```
7. Möglicherweise müssen Sie einen Exit für die automatische Kanaldefinition (CHADEXIT) im Clustersenderwarteschlangenmanager definieren, wenn sich die Warteschlangenmanager auf unterschiedlichen Plattformen befinden.

   Verwenden Sie den Exit für die automatische Kanaldefinition, um das Attribut SecurityExit auf ein geeignetes Format für die Zielplattform zu setzen.
8. Implementieren und konfigurieren Sie den Sicherheitsexit.

   z/OS z/OS

   Das Lademodul für Sicherheitsexits muss sich in der Datei befinden, die in der Anweisung CSQXLIB DD des Kanalinitiatoradressspeicherbereichs angegeben ist.

ALW Systeme mit AIX, Linux, and Windows

- Die Dynamic Link Library des Sicherheitsexits muss sich in dem Pfad befinden, der im Attribut SCYEXIT der Kanaldefinition angegeben ist.
- Die dynamische Link-Bibliothek für die automatische Kanaldefinition muss sich in dem Pfad befinden, der im Attribut CHADEXIT der Warteschlangenmanagerdefinition angegeben ist.

Unerwünschte WS-Manager zum Verlassen eines Clusters

Erzwingen Sie einen unerwünschten Warteschlangenmanager, einen Cluster zu verlassen, indem Sie den Befehl RESET CLUSTER an einem vollständigen WS-Manager-Repository absetzen.

Informationen zu diesem Vorgang

Sie können einen nicht erwünschten WS-Manager erzwingen, um einen Cluster zu verlassen. Wenn zum Beispiel ein Warteschlangenmanager gelöscht wird, dessen Clusterempfängerkanäle jedoch noch für den Cluster definiert sind. Vielleicht wollen Sie aufräumen.

Nur vollständige WS-Manager-Repositorys sind berechtigt, einen Warteschlangenmanager aus einem Cluster auszuschließen.

Anmerkung: Obwohl der Befehl RESET CLUSTER zwangsweise einen WS-Manager aus einem Cluster entfernt, verhindert die Verwendung von RESET CLUSTER durch sich selbst nicht, dass der WS-Manager später wieder in den Cluster einfügt. Führen Sie die in „Verhindern, dass WS-Manager in einen Cluster" auf Seite 512 beschriebenen Schritte aus, um sicherzustellen, dass der WS-Manager nicht wieder in den Cluster aufgenommen wird.

Gehen Sie wie folgt vor, um den WS-Manager OSLO aus dem Cluster NORWAY auszuwerfen:

Vorgehensweise

1. Geben Sie in einem vollständigen Repository-WS-Manager den folgenden Befehl aus:

```
RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
```

2. Alternativ können Sie die QMID anstelle von QMNAME in dem Befehl verwenden:

```
RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
```

Anmerkung: QMID ist eine Zeichenfolge, daher sollte der Wert von qmid in einfache Anführungszeichen eingeschlossen werden, z. B. QMID('FR01_2019-07-15_14.42.42').

Ergebnisse

Der Warteschlangenmanager, der entfernt wird, ändert sich nicht; seine lokalen Clusterdefinitionen zeigen, dass er im Cluster enthalten ist. Die Definitionen in allen anderen Warteschlangenmanagern zeigen sie nicht im Cluster an.

Verhindern, dass Warteschlangenmanager Nachrichten empfangen

Sie können verhindern, dass ein Cluster-WS-Manager Nachrichten empfängt, die er mit Exitprogrammen nicht empfangen kann.

Informationen zu diesem Vorgang

Es ist schwierig, einen Warteschlangenmanager zu stoppen, der aus der Definition einer Warteschlange Mitglied eines Clusters ist. Es besteht die Gefahr, dass ein Schurkenwarteschlangenmanager in einen Cluster aufgenommen wird und eine eigene Instanz einer der Warteschlangen im Cluster definiert. Es kann jetzt Nachrichten empfangen, die nicht berechtigt sind, zu empfangen. Um zu verhindern, dass ein

Warteschlangenmanager Nachrichten empfängt, verwenden Sie eine der folgenden Optionen, die in der Prozedur angegeben sind.

Prozedur

- Ein Kanalexitprogramm auf jedem Clustersenderkanal. Das Exitprogramm verwendet den Verbindungsnamen, um die Eignung des Zielwarteschlangenmanagers zu ermitteln, an den die Nachrichten gesendet werden sollen.
- Ein Exitprogramm für Clusterauslastung, das die Zieldatensätze verwendet, um die Eignung der Zielwarteschlange und des Warteschlangenmanagers zu ermitteln, an die die Nachrichten gesendet werden sollen.

SSL/TLS und Cluster

Wenn Sie TLS für Cluster konfigurieren, müssen Sie beachten, dass eine CLUSRCVR-Kanaldefinition an andere WS-Manager als automatisch definierter CLUSSDR-Kanal weitergegeben wird. Wenn ein CLUSRCVR-Kanal TLS verwendet, müssen Sie TLS auf allen Warteschlangenmanagern konfigurieren, die mit dem Kanal kommunizieren.

Weitere Informationen zu TLS finden Sie unter „TLS-Sicherheitsprotokolle in IBM MQ" auf Seite 26. Die Empfehlung gibt es im Allgemeinen für Clusterkanäle, aber Sie können die folgenden Hinweise beachten:

In einem IBM MQ-Cluster wird eine bestimmte CLUSRCVR-Kanaldefinition häufig an viele andere Warteschlangenmanager weitergegeben und dort in einen automatisch definierten Clustersenderkanal CLUSSDR umgewandelt. Anschließend wird die automatisch definierte CLUSSDR verwendet, um einen Kanal zum CLUSRCVR zu starten. Wenn der CLUSRCVR für die TLS-Konnektivität konfiguriert ist, gelten die folgenden Hinweise:

- Alle WS-Manager, die mit diesem CLUSRCVR kommunizieren wollen, müssen Zugriff auf die TLS-Unterstützung haben. Diese TLS-Bereitstellung muss die CipherSpec für den Kanal unterstützen.
- Die verschiedenen Warteschlangenmanager, an die die automatisch definierten Clustersenderkanäle weitergegeben wurden, haben jeweils einen anderen definierten Namen zugeordnet. Wenn die Peer-Prüfung für den registrierten Namen auf dem CLUSRCVR verwendet werden soll, muss sie so konfiguriert werden, dass alle definierten Namen, die empfangen werden können, erfolgreich abgeglichen werden.

  Nehmen wir zum Beispiel an, dass alle Warteschlangenmanager, die Clustersenderkanäle enthalten, die eine Verbindung zu einem bestimmten CLUSRCVR herstellen, Zertifikate zugeordnet haben. Nehmen wir außerdem an, dass in allen diesen Zertifikaten der definierte Name als Land 'Großbritannien', als Unternehmen 'IBM' und als Unternehmenseinheit 'IBM MQ Development' definiert ist. Alle haben allgemeine Namen im Format DEVT.QMnnn, wobei nnn für einen numerischen Wert steht.

  In diesem Fall kann ein SSLPEER -Wert von C=UK, O=IBM, OU=IBM MQ Development, CN=DEVT.QM* auf dem CLUSRCVR alle erforderlichen Clustersenderkanäle erfolgreich verbinden, verhindert jedoch, dass unerwünschte Clustersenderkanäle eine Verbindung herstellen.
- Wenn angepasste CipherSpec-Zeichenfolgen verwendet werden, müssen Sie beachten, dass die angepassten Zeichenfolgeformate nicht auf allen Plattformen zulässig sind. Ein Beispiel dafür ist, dass die CipherSpec -Zeichenfolge RC4_SHA_US einen Wert von 05 unter IBM i hat, aber keine gültige Spezifikation auf AIX, Linux, and Windows-Systemen darstellt. Wenn also angepasste SSLCIPH -Parameter auf einem CLUSRCVR verwendet werden, sollten sich alle resultierenden automatisch definierten Clustersenderkanäle auf Plattformen befinden, auf denen die zugrunde liegende TLS-Unterstützung diese CipherSpec implementiert und auf der sie mit dem angepassten Wert angegeben werden kann. Wenn Sie keinen Wert für den Parameter SSLCIPH auswählen können, der im gesamten Cluster verstanden wird, benötigen Sie einen Exit für die automatische Kanaldefinition, um ihn in etwas zu ändern, das die verwendeten Plattformen verstehen. Verwenden Sie die textuellen CipherSpec -Zeichenfolgen wo möglich (z. B. TLS_RSA_WITH_AES_128_CBC_SHA).

Ein Parameter SSLCRLNL gilt für einen einzelnen WS-Manager und wird nicht an andere Warteschlangenmanager innerhalb eines Clusters weitergegeben.

Upgrade für Cluster-WS-Manager und -Kanäle auf SSL/TLS durchführen

Führen Sie ein Upgrade der Clusterkanäle auf einmal durch, und ändern Sie alle CLUSRCVR -Kanäle vor den CLUSSDR -Kanälen.

Vorbereitende Schritte

Berücksichtigen Sie die folgenden Überlegungen, da diese Auswirkungen auf die Auswahl von CipherSpec für einen Cluster haben können:

- Einige CipherSpecs sind auf allen Plattformen nicht verfügbar. Wählen Sie eine CipherSpec aus, die von allen Warteschlangenmanagern im Cluster unterstützt wird.
- Einige CipherSpecs sind neu im aktuellen IBM MQ-Release und werden in älteren Releases nicht unterstützt. Ein Cluster mit WS-Managern, die in verschiedenen MQ-Releases ausgeführt werden, kann nur die CipherSpecs verwenden, die von jedem Release unterstützt werden.

  Wenn Sie eine neue CipherSpec in einem Cluster verwenden möchten, müssen Sie zuerst alle Cluster-WS-Manager auf das aktuelle Release migrieren.
- Für einige CipherSpecs ist ein bestimmter Typ des zu verwendenden digitalen Zertifikats erforderlich, insbesondere solche, die die Elliptic Curve Cryptography verwenden.

  Achtung: Es ist nicht möglich, eine Kombination aus Zertifikaten, die mit Elliptic Curve und RSA signiert wurden, auf Warteschlangenmanagern zu verwenden, die als Teil eines Clusters miteinander verknüpft werden sollen.

  Warteschlangenmanager in einem Cluster müssen entweder mit RSA signierte Zertifikate oder mit EC signierte Zertifikate verwenden und nicht eine Kombination aus beiden.

  Weitere Informationen finden Sie unter „Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ" auf Seite 49.

Aktualisieren Sie alle Warteschlangenmanager im Cluster auf IBM MQ V8 oder höher, wenn sie sich nicht bereits auf diesen Ebenen befinden. Verteilen Sie die Zertifikate und Schlüssel so, dass TLS von jedem von ihnen funktioniert.

Wenn Sie eine der Alias-CipherSpecs (ANY_TLS13, ANY_TLS13_OR_HIGHER, ANY_TLS12, ANY_TLS12_OR_HIGHER, usw) aktualisieren oder verwenden möchten, müssen Sie alle IBM MQ for Multiplatforms-Warteschlangenmanager im Cluster auf IBM MQ 9.1.4 oder höher V 9.2.0 und alle IBM MQ for z/OS-Warteschlangenmanager im Cluster auf IBM MQ for z/OS 9.2.0 oder höher aktualisieren.

Informationen zu diesem Vorgang

Ändern Sie die CLUSRCVR -Kanäle vor den CLUSSDR -Kanälen.

Vorgehensweise

1. Schalten Sie die CLUSRCVR -Kanäle in einer beliebigen Reihenfolge in TLS um, ändern Sie jeweils einen CLUSRCVR und lassen Sie die Änderungen an den Änderungen durch den Cluster fließen, bevor Sie die nächste ändern.

   Wichtig: Stellen Sie sicher, dass Sie den Reverse-Pfad erst ändern, wenn die Änderungen für den aktuellen Kanal im gesamten Cluster verteilt wurden.
2. Optional: Schalten Sie alle manuellen CLUSSDR -Kanäle in TLS um.

   Dies wirkt sich nicht auf die Operation des Clusters aus, es sei denn, Sie verwenden den Befehl REFRESH CLUSTER mit der Option REPOS(YES).

   Anmerkung: Bei großen Clustern kann die Verwendung des Befehls **REFRESH CLUSTER** während der Ausführung des Clusters und danach in 27-Tage-Intervallen, wenn die Clusterobjekte automatisch Statusaktualisierungen an alle interessierten Warteschlangenmanager senden, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.

3. Verwenden Sie den Befehl DISPLAY CLUSQMGR , um sicherzustellen, dass die neue Sicherheitskonfiguration über den gesamten Cluster weitergegeben wurde.
4. Starten Sie die Kanäle erneut, um TLS zu verwenden, und führen Sie REFRESH SECURITY (SSL) aus.

Zugehörige Konzepte

„CipherSpecs aktivieren" auf Seite 448
Aktivieren Sie eine CipherSpec mit dem Parameter **SSLCIPH** im MQSC-Befehl **DEFINE CHANNEL** oder **ALTER CHANNEL**.

„Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ" auf Seite 49
Dieser Abschnitt enthält Informationen dazu, wie Sie die richtigen CipherSpecs und digitalen Zertifikate für Ihre Sicherheitsrichtlinie auswählen. Dazu wird die Beziehung zwischen CipherSpecs und digitalen Zertifikaten in IBM MQ erläutert.

Zugehörige Informationen

Clustering: Best Practices für REFRESH CLUSTER verwenden

SSL/TLS auf Cluster-WS-Managern und -Kanälen inaktivieren

Um TLS zu inaktivieren, setzen Sie den Parameter SSLCIPH auf `' '`. Inaktivieren Sie TLS einzeln auf den Clusterkanälen, und ändern Sie alle Clusterempfängerkanäle vor den Clustersenderkanälen.

Informationen zu diesem Vorgang

Ändern Sie einen Clusterempfängerkanal zu einem Zeitpunkt und lassen Sie die Änderungen an den Änderungen durch den Cluster fließen, bevor Sie den nächsten ändern.

Wichtig: Stellen Sie sicher, dass Sie den Reverse-Pfad erst ändern, wenn die Änderungen für den aktuellen Kanal im gesamten Cluster verteilt wurden.

Vorgehensweise

1. Setzen Sie den Wert des Parameters SSLCIPH auf `' '`, eine leere Zeichenfolge in einem einfachen Anführungszeichen IBM i oder *NONE unter IBM i .

   Sie können TLS auf den Clusterempfängerkanälen in jeder beliebigen Reihenfolge abschalten.

   Beachten Sie, dass die Änderungen in die entgegengesetzte Richtung über Kanäle fließen, auf denen Sie TLS aktiv lassen.
2. Überprüfen Sie, ob der neue Wert in allen anderen Queue Managern über den Befehl **DISPLAY CLUSQMGR(*)** ALL widergespiegelt wird.
3. Schalten Sie TLS auf allen manuellen Clustersenderkanälen aus.

   Dies wirkt sich nicht auf die Operation des Clusters aus, es sei denn, Sie verwenden den Befehl **REFRESH CLUSTER** mit der Option REPOS(YES) .

   Bei großen Clustern kann die Verwendung des Befehls **REFRESH CLUSTER** den Cluster unterbrechen, während er in Bearbeitung ist, und danach in regelmäßigen Intervallen, wenn die Clusterobjekte automatisch Statusaktualisierungen an alle interessierten Warteschlangenmanager senden. Weitere Informationen hierzu finden Sie im Abschnitt In einem großen Cluster aktualisieren kann die Leistung und Verfügbarkeit des Clusters beeinträchtigen .
4. Stoppen Sie die Clustersenderkanäle und starten Sie sie erneut.

Publish/Subscribe-Sicherheit

Die Komponenten und Interaktionen, die an Publish/Subscribe beteiligt sind, werden als Einführung in die ausführlicheren Erläuterungen und Beispiele beschrieben, die folgen.

Es gibt eine Reihe von Komponenten, die beim Veröffentlichen und Subskribieren eines Themas beteiligt sind. Einige der Sicherheitsbeziehungen zwischen ihnen werden in Abbildung 22 auf Seite 518 dargestellt und im folgenden Beispiel beschrieben.

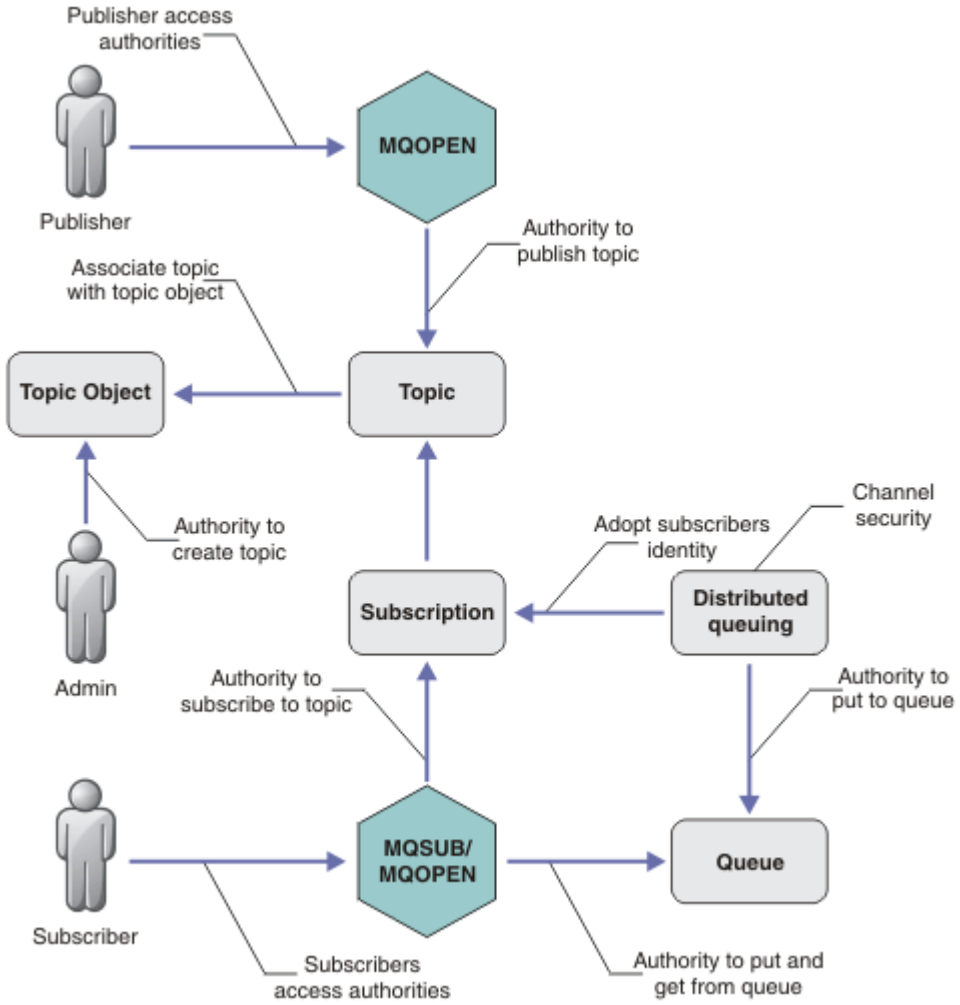


Abbildung 22. Publish/Subscribe-Sicherheitsbeziehungen zwischen verschiedenen Komponenten

Themen

Themen werden durch Themenzeichenfolgen identifiziert und sind in der Regel in Baumstrukturen organisiert, siehe Themenbäume. Sie müssen ein Thema einem Themenobjekt zuordnen, um den Zugriff auf das Thema zu steuern. In „Thema Sicherheitsmodell" auf Seite 520 wird erläutert, wie Sie Themen unter Verwendung von Themenobjekten schützen.

Verwaltungsthemenobjekte

Sie können steuern, wer zu welchem Zweck Zugriff auf ein Thema hat, indem Sie den Befehl **setmqaut** mit einer Liste von Verwaltungsthemenobjekten verwenden. Weitere Informationen finden Sie in den Beispielen, „Einem Benutzer den Zugriff auf die Subskription eines Themas gewähren" auf Seite 525 und „Einem Benutzer den Zugriff auf die Veröffentlichung in einem Thema gewähren" auf Seite 533. z/OS Informationen zum Steuern des Zugriffs auf Themenobjekte unter z/OS finden Sie im Abschnitt Profile für Themensicherheit.

Abonnements

Subskribieren Sie ein oder mehrere Themen, indem Sie eine Subskription erstellen, die eine Themenzeichenfolge bereitstellt, die Platzhalterzeichen enthalten kann, die mit den Themenzeichenfolgen von Veröffentlichungen übereinstimmen. Weitere Einzelheiten finden Sie unter:

Subskription mit einem Themenobjekt

„Abonnieren des Themenobjektnamens" auf Seite 521

Subskription mit einem Thema
„Abonnieren mit einer Themenzeichenfolge, in der der Themenknoten nicht vorhanden ist" auf Seite 522

Subskription unter Verwendung eines Themas mit Platzhalterzeichen
„Subskription mit einer Themenzeichenfolge, die Platzhalterzeichen enthält" auf Seite 523

Eine Subskription enthält Informationen über die Identität des Subskribenten und die Identität der Zielwarteschlange, in die die Veröffentlichungen gestellt werden sollen. Sie enthält außerdem Informationen darüber, wie die Veröffentlichung in die Zielwarteschlange gestellt werden soll.

Neben der Definition, welche Subskribenten die Berechtigung zum Subskribieren bestimmter Themen haben, können Sie die Subskriptionen einschränken, die von einem einzelnen Subskribenten verwendet werden. Sie können auch steuern, welche Informationen über den Subskribenten vom Warteschlangenmanager verwendet werden, wenn Veröffentlichungen in die Zielwarteschlange gestellt werden. Siehe „Subskriptionssicherheit" auf Seite 538.

Warteschlangen

Die Zielwarteschlange ist eine wichtige Warteschlange für die Sicherung. Es ist lokal für den Subskribenten, und die Veröffentlichungen, die mit der Subskription übereinstimmen, werden auf diese gestellt. Sie müssen den Zugriff auf die Zielwarteschlange aus zwei Perspektiven in Betracht ziehen:

1. Veröffentlichung einer Veröffentlichung in die Zielwarteschlange.
2. Die Veröffentlichung wird aus der Zielwarteschlange abgerufen.

Der Warteschlangenmanager stellt eine Veröffentlichung unter Verwendung einer vom Subskribenten zur Verfügung gestellten Identität in die Zielwarteschlange. Der Subskribent oder ein Programm, das die Task zum Abrufen von Veröffentlichungen delegiert hat, nimmt Nachrichten aus der Warteschlange ab. Siehe „Berechtigung für Zielwarteschlangen" auf Seite 523.

Es gibt keine Themenobjektaliasnamen, aber Sie können eine Aliaswarteschlange als Aliasname für ein Themenobjekt verwenden. Wenn Sie dies tun, und die Berechtigung zur Verwendung des Themas für Publish/Subscribe überprüfen, prüft der Warteschlangenmanager die Berechtigung zur Verwendung der Warteschlange.

„Publish/Subscribe-Sicherheit zwischen Warteschlangenmanagern" auf Seite 540

Ihre Berechtigung zum Veröffentlichen oder Subskribieren eines Themas wird auf dem lokalen WS-Manager unter Verwendung lokaler Identitäten und Berechtigungen überprüft. Die Autorisierung hängt nicht davon ab, ob das Thema definiert ist oder nicht, und nicht, wo es definiert ist. Daher müssen Sie die Topic-Berechtigung für jeden Warteschlangenmanager in einem Cluster ausführen, wenn die Cluster-Themen verwendet werden.

Anmerkung: Das Sicherheitsmodell für Themen unterscheidet sich von dem Sicherheitsmodell für Warteschlangen. Sie können dasselbe Ergebnis für Warteschlangen erzielen, indem Sie einen Warteschlangenalias für jede Clusterwarteschlange lokal definieren.

WS-Manager tauschen Subskriptionen in einem Cluster aus. In den meisten IBM MQ-Clusterkonfigurationen werden Kanäle mit PUTAUT=DEF konfiguriert, um Nachrichten mithilfe der Berechtigung des Kanalprozesses in Zielwarteschlange zu stellen. Sie können die Kanalkonfiguration so ändern, dass sie PUTAUT=CTX verwendet, um zu verlangen, dass der subskribierende Benutzer über die Berechtigung verfügt, eine Subskription an einen anderen WS-Manager in einem Cluster weiterzugeben.

In „Publish/Subscribe-Sicherheit zwischen Warteschlangenmanagern" auf Seite 540 wird beschrieben, wie Sie Ihre Kanaldefinitionen ändern, um zu steuern, wer die Subskriptionen an andere Server im Cluster weitergeben darf.

Berechtigung

Sie können die Berechtigung für Themenobjekte wie Warteschlangen und andere Objekte anwenden. Es gibt drei Berechtigungsoperationen, `pub` , `sub` und `resume` , die Sie nur auf Themen anwenden können. Die Details sind im Abschnitt Berechtigungen für verschiedene Objektarten angeben beschrieben.

Funktionsaufrufe

In Publish/Subscribe-Programmen wie in in der Warteschlange befindlichen Programmen werden Berechtigungsprüfungen durchgeführt, wenn Objekte geöffnet, erstellt, geändert oder gelöscht werden. Es werden keine Prüfungen durchgeführt, wenn MQPUT -oder MQGET MQI-Aufrufe zum Einreihen und Abrufen von Veröffentlichungen ausgeführt werden.

Um ein Thema zu veröffentlichen, führen Sie eine MQOPEN für das Thema aus, die die Berechtigungsprüfungen durchführt. Veröffentlichen Sie Nachrichten im Topic-Handle mit dem Befehl MQPUT , der keine Berechtigungsprüfungen durchführt.

Um ein Thema zu subskribieren, führen Sie in der Regel einen MQSUB -Befehl aus, um die Subskription zu erstellen oder wiederaufzunehmen und um die Zielwarteschlange zum Empfangen von Veröffentlichungen zu öffnen. Alternativ können Sie eine separate MQOPEN ausführen, um die Zielwarteschlange zu öffnen, und anschließend die MQSUB ausführen, um die Subskription zu erstellen bzw. fortzusetzen.

Whichever-Aufrufe, die Sie verwenden, prüft der Warteschlangenmanager, ob Sie das Thema subskribieren können, und die resultierenden Veröffentlichungen aus der Zielwarteschlange abrufen. Wenn die Zielwarteschlange nicht verwaltet wird, werden Berechtigungsprüfungen durchgeführt, die der Warteschlangenmanager in der Lage ist, Veröffentlichungen in die Zielwarteschlange zu stellen. Sie verwendet die Identität, die sie aus einer übereinstimmenden Subskription übernommen hat. Es wird davon ausgegangen, dass der Warteschlangenmanager immer in der Lage ist, Veröffentlichungen in die Warteschlangen des verwalteten Ziels zu stellen.

Rollen

Benutzer sind an der Ausführung von Publish/Subscribe-Anwendungen in vier Rollen beteiligt:

1. Bereitsteller
2. Subskribent
3. Topic-Administrator
4. IBM MQ Administrator-Mitglied der Gruppe mqm

Definieren Sie Gruppen mit den entsprechenden Berechtigungen, die den Rollen für die Veröffentlichung, Subskriptionssubskriptionsgruppe und die Topic-Verwaltung entsprechen. Anschließend können Sie Principals diesen Gruppen zuordnen, die sie berechtigen, bestimmte Publish/Subscribe-Tasks auszuführen.

Darüber hinaus müssen Sie die Berechtigungen der Verwaltungsoperationen auf den Administrator der Warteschlangen und Kanäle, die für das Verschieben von Veröffentlichungen und Subskriptionen verantwortlich sind, erweitern.

Thema Sicherheitsmodell

Nur definierte Themenobjekte können zugeordnete Sicherheitsattribute aufweisen. Eine Beschreibung der Themenobjekte finden Sie unter Verwaltungsthemenobjekte. Die Sicherheitsattribute geben an, ob eine angegebene Benutzer-ID oder Sicherheitsgruppe berechtigt ist, eine Subskription oder eine Veröffentlichungsoperation für jedes Themenobjekt auszuführen.

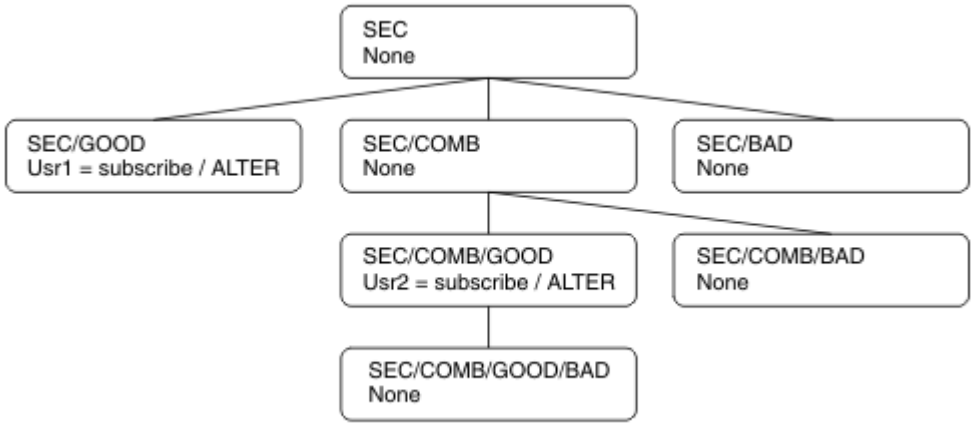
Die Sicherheitsattribute sind dem entsprechenden Verwaltungsknoten in der Themenstruktur zugeordnet. Wenn eine Berechtigungs-Prüfung für eine bestimmte Benutzer-ID während einer Subskription-oder Veröffentlichungsoperation durchgeführt wird, basiert die erteilte Berechtigung auf den Sicherheitsattributen des zugeordneten Themenbaumknotens.

Die Sicherheitsattribute sind eine Zugriffssteuerungsliste, die angibt, welche Berechtigung eine bestimmte Betriebssystembenutzer-ID oder Sicherheitsgruppe für das Themenobjekt hat.

Betrachten Sie das folgende Beispiel, in dem die Themenobjekte mit den Sicherheitsattributen oder den angezeigten Berechtigungen definiert wurden:

Tabelle 84. Beispiel-Topic-Objektberechtigungen			
Themenname	**Themenzeichen-folge**	**Berechtigungen - nicht z/OS**	**z/OS-Berechtigungen**
SECROOT	SEC	--	--
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD
SECBAD	SEC/BAD	--	-- HLQ.SUBSCRIBE.SECBAD
SECCOMB	SEC/COMB	--	-- HLQ.SUBSCRIBE.SECCOMB
SECCOMBB	SEC/COMB/ GOOD/BAD	--	-- HLQ.SUBSCRIBE.SECCOMBB
SECCOMBG	SEC/COMB/GOOD	usr2+subscribe	ALTER HLQ.SUBSCRIBE.SECCOMBG
SECCOMBN	SEC/COMB/BAD	--	-- HLQ.SUBSCRIBE.SECCOMBN

Die Themenstruktur mit den zugehörigen Sicherheitsattributen an den einzelnen Knoten kann wie folgt dargestellt werden:



In den aufgeführten Beispielen werden die folgenden Berechtigungen erteilt:

- Auf dem Stammknoten der Baumstruktur /SEC hat kein Benutzer die Berechtigung für diesen Knoten.
- usr1 hat die Berechtigung zur Subskription des Objekts /SEC/GOOD
- usr2 hat die Berechtigung zur Subskription des Objekts /SEC/COMB/GOOD

Abonnieren des Themenobjektnamens

Wenn Sie ein Themenobjekt subskribieren, indem Sie den Namen MQCHAR48 angeben, befindet sich der entsprechende Knoten in der Themenstruktur. Wenn die Sicherheitsattribute, die dem Knoten zugeordnet sind, angeben, dass der Benutzer über die Berechtigung zum Subskribieren verfügt, wird der Zugriff erteilt.

Wenn dem Benutzer kein Zugriff gewährt wird, bestimmt der übergeordnete Knoten in der Baumstruktur, ob der Benutzer über die Berechtigung zum Subskribieren auf der Ebene des übergeordneten Knotens verfügt. Ist dies der Fall, wird der Zugriff gewährt. Ist dies nicht der Fall, wird der übergeordnete Knoten dieses Knotens berücksichtigt. Die Rekursion wird so lange fortgesetzt, bis ein Knoten gefunden wird, der dem Benutzer die Subskriptionsberechtigung erteilt. Die Rekursion wird gestoppt, wenn der Rootknoten ohne Berechtigung als erteilt betrachtet wird. In letzterem Fall wird der Zugriff verweigert.

Kurz, wenn ein Knoten im Pfad berechtigt ist, diesen Benutzer oder die Anwendung zu subskribieren, kann der Subskribent an diesem Knoten oder an einer beliebigen Stelle unterhalb dieses Knotens in der Themenstruktur subskribieren.

Der Stammknoten im Beispiel ist SEC.

Dem Benutzer wird die Subskriptionsberechtigung erteilt, wenn die Zugriffssteuerungsliste angibt, dass die Benutzer-ID selbst über die Berechtigung verfügt oder dass eine Sicherheitsgruppe des Betriebssystems, zu der die Benutzer-ID gehört, die Berechtigung hat.

So, zum Beispiel:

- Wenn `usr1` versucht, eine Subskription unter Verwendung einer Themenzeichenfolge von `SEC/GOOD` zu verwenden, ist die Subskription zulässig, da die Benutzer-ID Zugriff auf den Knoten hat, der diesem Thema zugeordnet ist. Wenn `usr1` jedoch versucht hat, die Themenzeichenfolge `SEC/COMB/GOOD` zu subskribieren, wäre die Subskription nicht zulässig, da die Benutzer-ID nicht über Zugriff auf den zugehörigen Knoten verfügt.
- Wenn `usr2` versucht, eine Subskription zu erhalten, wird die Subskription über eine Themenzeichenfolge von `SEC/COMB/GOOD` zugelassen, da die Benutzer-ID über Zugriff auf den Knoten verfügt, der dem Thema zugeordnet ist. Wenn `usr2` jedoch versucht hat, `SEC/GOOD` zu subskribieren, ist die Subskription nicht zulässig, da die Benutzer-ID nicht über Zugriff auf den zugehörigen Knoten verfügt.
- Wenn `usr2` versucht, eine Subskription unter Verwendung einer Themenzeichenfolge von `SEC/COMB/GOOD/BAD` zu verwenden, kann die Subskription zugelassen werden, da die Benutzer-ID Zugriff auf den übergeordneten Knoten `SEC/COMB/GOOD` hat.
- Wenn `usr1` oder `usr2` versucht, eine Subskription unter Verwendung einer Themenzeichenfolge von `/SEC/COMB/BAD` zu verwenden, ist dies nicht zulässig, da sie keinen Zugriff auf den zugehörigen Themenknoten haben oder die übergeordneten Knoten dieses Themas haben.

Eine Subskriptionsoperation, die den Namen eines Themenobjekts angibt, das nicht vorhanden ist, führt zu einem Fehler MQRC_UNKNOWN_OBJECT_NAME.

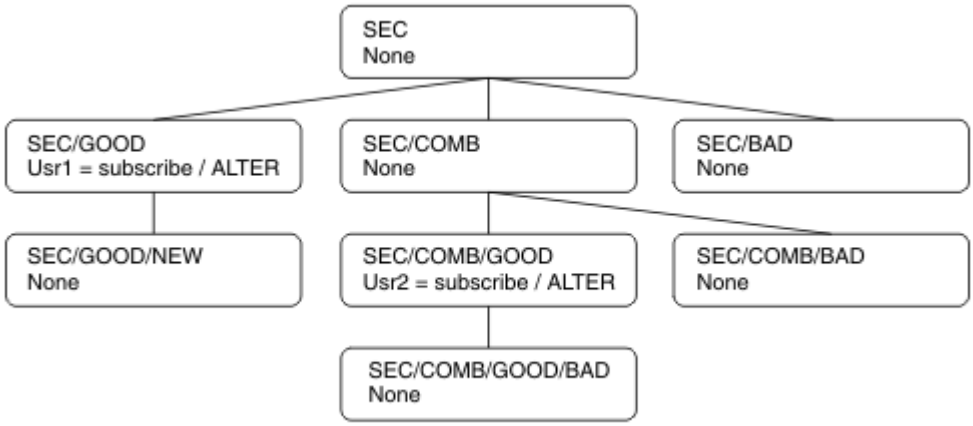
Abonnieren mit einer Themenzeichenfolge, in der der Themenknoten vorhanden ist

Das Verhalten ist dasselbe wie bei der Angabe des Themas durch den Namen des MQCHAR48-Objekts.

Abonnieren mit einer Themenzeichenfolge, in der der Themenknoten nicht vorhanden ist

Betrachten Sie den Fall einer Anwendung, die subskribiert ist, und geben Sie eine Themenzeichenfolge an, die einen Themenknoten darstellt, der derzeit nicht in der Themenstruktur vorhanden ist. Die Berechtigschecks werden wie im vorherigen Abschnitt beschrieben ausgeführt. Die Prüfung beginnt mit dem übergeordneten Knoten des Elements, das durch die Themenzeichenfolge dargestellt wird. Wenn die Berechtigung erteilt wird, wird in der Themenstruktur ein neuer Knoten erstellt, der die Themenzeichenfolge darstellt.

`usr1` versucht z. B., ein Thema `SEC/GOOD/NEW` zu subskribieren. Die Berechtigung wird erteilt, da `usr1` Zugriff auf den übergeordneten Knoten `SEC/GOOD` hat. Es wird ein neuer Themenknoten in der Baumstruktur erstellt, wie im folgenden Diagramm dargestellt. Der neue Themenknoten ist kein Themenobjekt, dem keine Sicherheitsattribute zugeordnet sind. Die Attribute werden von seinem übergeordneten Knoten übernommen.



Subskription mit einer Themenzeichenfolge, die Platzhalterzeichen enthält

Berücksichtigen Sie den Fall, dass eine Themenzeichenfolge subskribiert wird, die ein Platzhalterzeichen enthält. Die Berechtigungs-Prüfung wird für den Knoten in der Themenstruktur durchgeführt, der mit dem vollständig qualifizierten Teil der Themenzeichenfolge übereinstimmt.

Wenn eine Anwendung also eine Subskription für `SEC/COMB/GOOD/*` subskribiert, wird eine Berechtigungs-Prüfung wie in den vorherigen zwei Abschnitten auf dem Knoten `SEC/COMB/GOOD` in der Themenstruktur ausgeführt.

Wenn eine Anwendung `SEC/COMB/*/GOOD` subskribieren muss, wird auch eine Berechtigungs-Prüfung auf dem Knoten `SEC/COMB` durchgeführt.

Berechtigung für Zielwarteschlangen

Beim Abonnieren eines Themas ist einer der Parameter die Kennung `hobj` einer Warteschlange, die für die Ausgabe geöffnet wurde, um die Veröffentlichungen zu empfangen.

Wenn `hobj` nicht angegeben ist, aber leer ist, wird eine verwaltete Warteschlange erstellt, wenn die folgenden Bedingungen gelten:

- Die Option `MQSO_MANAGED` wurde angegeben.
- Die Subskription ist nicht vorhanden.
- Erstellen ist angegeben.

Wenn `hobj` leer ist und Sie ein vorhandenes Abonnement ändern oder wieder aufnehmen, kann die zuvor angegebene Zielwarteschlange entweder verwaltet oder nicht verwaltet werden.

Die Anwendung oder der Benutzer, die bzw. der die `MQSUB` -Anforderung stellt, muss über die Berechtigung zum Einreihen von Nachrichten in die angegebene Zielwarteschlange verfügen. In der Tat muss die Berechtigung zum Einreihen von Nachrichten in diese Warteschlange vorliegen. Die Berechtigungsprüfung folgt den vorhandenen Regeln für die Warteschlangensicherheitsüberprüfung.

Die Sicherheitsprüfung umfasst alternative Benutzer-ID und Kontextsicherheitsüberprüfungen, falls erforderlich. Damit Sie einen der Identitätskontextfelder festlegen können, müssen Sie die Option `MQSO_SET_IDENTITY_CONTEXT` sowie die Option `MQSO_CREATE` oder `MQSO_ALTER` angeben. Sie können keine Identitätskontextfelder in einer `MQSO_RESUME` -Anforderung festlegen.

Wenn es sich bei dem Ziel um eine verwaltete Warteschlange handelt, werden keine Sicherheitsprüfungen für das verwaltete Ziel durchgeführt. Wenn Sie ein Thema subskribieren dürfen, wird davon ausgegangen, dass Sie verwaltete Ziele verwenden können.

Veröffentlichung unter Verwendung des Topic-Namens oder der Themenzeichenfolge, in der der Themenknoten vorhanden ist

Das Sicherheitsmodell für die Veröffentlichung ist mit dem für das Subskribieren identisch, mit Ausnahme von Platzhaltern. Veröffentlichungen enthalten keine Platzhalterzeichen. Daher gibt es keinen Fall für eine Themenzeichenfolge, die Platzhalterzeichen enthält.

Die zu veröffentlichungs-und subskriptionsspezifischen Berechtigungen sind unterschiedlich Ein Benutzer oder eine Gruppe kann die Berechtigung haben, einen Benutzer zu machen, ohne dass er die Möglichkeit hat, die andere zu tun.

Wenn die Veröffentlichung in einem Themenobjekt erfolgt, indem entweder der Name des MQCHAR48-Namens oder die Themenzeichenfolge angegeben wird, befindet sich der entsprechende Knoten in der Themenstruktur. Wenn die Sicherheitsattribute, die dem Themenknoten zugeordnet sind, angeben, dass der Benutzer über die Berechtigung zum Publizieren verfügt, wird der Zugriff erteilt.

Wenn der Zugriff nicht erteilt wird, bestimmt der übergeordnete Knoten in der Baumstruktur, ob der Benutzer über die Berechtigung zum Veröffentlichen auf dieser Ebene verfügt. Ist dies der Fall, wird der Zugriff gewährt. Ist dies nicht der Fall, wird die Rekursion so lange fortgesetzt, bis ein Knoten gefunden wird, der dem Benutzer die Veröffentlichungsberechtigung erteilt. Die Rekursion wird gestoppt, wenn der Rootknoten ohne Berechtigung als erteilt betrachtet wird. In letzterem Fall wird der Zugriff verweigert.

Kurz, wenn ein Knoten im Pfad berechtigt ist, die Veröffentlichung für diesen Benutzer oder die Anwendung zu veröffentlichen, darf der Publisher an diesem Knoten oder an einer beliebigen Stelle unterhalb dieses Knotens in der Themenstruktur veröffentlichen.

Veröffentlichung unter Verwendung des Topic-Namens oder der Themenzeichenfolge, in der der Themenknoten nicht vorhanden ist

Wie bei der Subskriptionsoperation, wenn eine Anwendung veröffentlicht, die eine Themenzeichenfolge angibt, die einen Themenknoten darstellt, der derzeit nicht in der Themenstruktur vorhanden ist, wird die Berechtigungs-Check-Operation beginnend mit dem übergeordneten Knoten des Knotens ausgeführt, der durch die Themenzeichenfolge dargestellt wird. Wenn die Berechtigung erteilt wird, wird in der Themenstruktur ein neuer Knoten erstellt, der die Themenzeichenfolge darstellt.

Veröffentlichung unter Verwendung einer Aliaswarteschlange, die in ein Themenobjekt aufgelöst wird

Wenn Sie die Veröffentlichung mithilfe einer Aliaswarteschlange veröffentlichen, die in ein Themenobjekt aufgelöst wird, erfolgt die Sicherheitsprüfung sowohl in der Aliaswarteschlange als auch in dem zugrunde liegenden Thema, in das sie aufgelöst wird.

Die Sicherheitsüberprüfung in der Aliaswarteschlange prüft, ob der Benutzer die Berechtigung zum Einlegen von Nachrichten in diese Aliaswarteschlange hat, und die Sicherheitsprüfung für das Thema prüft, ob der Benutzer in diesem Thema veröffentlichen kann. Wenn eine Aliaswarteschlange in eine andere Warteschlange aufgelöst wird, werden keine Prüfungen in der zugrunde liegenden Warteschlange durchgeführt. Die Berechtigungsprüfung wird für Themen und Warteschlangen unterschiedlich ausgeführt.

Eine Subskription schließen

Wenn Sie eine Subskription mit der Option MQCO_REMOVE_SUB schließen, wird eine zusätzliche Sicherheitsprüfung durchgeführt, wenn Sie die Subskription unter dieser Kennung nicht erstellt haben.

Es wird eine Sicherheitsprüfung durchgeführt, um sicherzustellen, dass Sie über die entsprechende Berechtigung verfügen, um dies zu tun, wenn die Aktion zum Entfernen der Subskription führt. Wenn die Sicherheitsattribute, die dem Themenknoten zugeordnet sind, angeben, dass der Benutzer über die entsprechende Berechtigung verfügt, wird der Zugriff erteilt. Ist dies nicht der Fall, wird der übergeordnete Knoten in der Baumstruktur berücksichtigt, um zu ermitteln, ob der Benutzer die Berechtigung zum Schließen der Subskription hat. Die Rekursion wird fortgesetzt, bis entweder die Berechtigung erteilt oder der Rootknoten erreicht ist.

Definieren, Ändern und Löschen einer Subskription

Es werden keine Sicherheitsprüfungen für Abonnements durchgeführt, wenn eine Subskription administrativ erstellt wird, anstatt eine MQSUB -API-Anforderung zu verwenden. Der Administrator hat diese Berechtigung bereits über den Befehl erteilt.

Es werden Sicherheitsprüfungen durchgeführt, um sicherzustellen, dass Veröffentlichungen in die Zielwarteschlange gestellt werden können, die der Subskription zugeordnet ist. Die Prüfungen werden auf dieselbe Weise wie für eine MQSUB -Anforderung ausgeführt.

Die Benutzer-ID, die für diese Sicherheitsprüfungen verwendet wird, hängt von dem Befehl ab, der ausgegeben wird. Wenn der Parameter **SUBUSER** angegeben wird, wirkt sich dies auf die Art und Weise aus, wie die Prüfung ausgeführt wird (siehe Tabelle 85 auf Seite 525):

Tabelle 85. Benutzer-IDs, die für Sicherheitsprüfungen für Befehle verwendet werden

Befehl	SUBUSER angegeben und leer	SUBUSER angegeben und abgeschlossen	SUBUSER nicht angegeben
	Administrator-ID verwenden		Verwenden. Sie die Benutzer-ID aus dem LIKE-Abonnement
	Administrator-ID verwenden		Verwenden.DE- Sie die Be-FAULT.SUB- nutzer-IDwenn diese aus derAngabe leer Subskripti-ist, verwen- on SYSTEMden Sie die Administrator-ID
	Administrator-ID verwenden		Verwenden. Sie die Benutzer-ID aus der vorhandenen Subskription

Die einzige Sicherheitsprüfung, die beim Löschen von Subskriptionen mit dem Befehl DELETE SUB ausgeführt wird, ist die Befehlssicherheitsüberprüfung.

Beispiel für eine Publish/Subscribe-Sicherheitskonfiguration

In diesem Abschnitt wird ein Szenario beschrieben, das die Zugriffssteuerung für Themen in einer Weise konfiguriert hat, die es ermöglicht, die Sicherheitssteuerung bei Bedarf anzuwenden.

Einem Benutzer den Zugriff auf die Subskription eines Themas gewähren

Dieses Thema ist die erste in einer Liste mit Tasks, in denen Sie erfahren, wie Sie Zugriff auf Themen von mehr als einem Benutzer erteilen können.

Informationen zu diesem Vorgang

Bei dieser Task wird davon ausgegangen, dass keine Verwaltungsthemenobjekte vorhanden sind und dass keine Profile für die Subskription oder Veröffentlichung definiert wurden. Die Anwendungen erstellen neue Subskriptionen, statt vorhandene zu summieren, und verwenden nur die Themenzeichenfolge.

Eine Anwendung kann eine Subskription erstellen, indem sie ein Themenobjekt oder eine Themenzeichenfolge oder eine Kombination aus beiden bereitstellt. Whichever Art und Weise, wie die Anwendung auswählt, ist die Wirkung, eine Subskription an einem bestimmten Punkt in der Themenstruktur zu erstellen. Wenn dieser Punkt in der Themenstruktur durch ein Verwaltungsthemenobjekt dargestellt wird, wird ein Sicherheitsprofil basierend auf dem Namen dieses Themenobjekts überprüft.

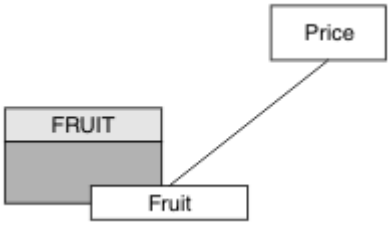


Abbildung 23. Zugriffsbeispiel für Themenobjektzugriff

Tabelle 86. Beispielthemenobjektzugriff

Thema	Subskriptionszugriff erforderlich	Themenobjekt
Preis	Kein Benutzer	--
Preis/>Obst	USER1	OBST

Definieren Sie ein neues Themenobjekt wie folgt:

Vorgehensweise

1. Setzen Sie den MQSC-Befehl `DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit')` ab.
2. Gehen Sie wie folgt vor:
   - z/OS **z/OS** :

     Erteilen Sie den Zugriff auf USER1 , um das Thema `"Price/Fruit"` zu subskribieren, indem Sie dem Benutzer Zugriff auf das `hlq.SUBSCRIBE.FRUIT` -Profil erteilen. Führen Sie dazu die folgenden RACF-Befehle aus:

     ```
     RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
     PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
     ```

   - Andere Plattformen:

     Erteilen Sie den Zugriff auf USER1 , um das Thema `"Price/Fruit"` zu abonnieren, indem Sie dem Benutzer Zugriff auf das Objekt FRUIT erteilen. Führen Sie dazu den Berechtigungsbefehl für die Plattform aus:

     ALW **AIX, Linux, and Windows-Systeme**

     ```
     setmqaut -t topic -n FRUIT -p USER1 +sub
     ```

IBM i

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
```

Ergebnisse

Wenn USER1 versucht, das Thema "Price/Fruit" zu abonnieren, wird das Ergebnis erfolgreich ausgeführt.

Wenn USER2 versucht, das Thema "Price/Fruit" subskribieren zu können, ist das Ergebnis mit einer MQRC_NOT_AUTHORIZED -Nachricht zusammen mit den folgenden Informationen fehlgeschlagen:

- z/OS Unter z/OS werden die folgenden Nachrichten in der Konsole angezeigt, die den vollständigen Sicherheitspfad über die Themenstruktur zeigen, die ausgeführt werden sollte:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW Auf anderen Plattformen ist das folgende Autorisierungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i Unter IBMi das folgende Berechtigungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

Beachten Sie, dass es sich hier um ein Beispiel für das, was Sie sehen, nicht um alle Felder handelt.

Einem Benutzer Zugriff gewähren, um ein Thema tiefer in der Baumstruktur zu subskribieren

Dieses Thema ist die zweite in einer Liste von Tasks, in denen Sie erfahren, wie Sie Zugriff auf Themen von mehr als einem Benutzer erteilen können.

Vorbereitende Schritte

In diesem Abschnitt wird die in „Einem Benutzer den Zugriff auf die Subskription eines Themas gewähren” auf Seite 525 beschriebene Konfiguration verwendet.

Informationen zu diesem Vorgang

Wenn der Punkt in der Themenstruktur, in dem die Anwendung die Subskription herstellt, nicht durch ein Verwaltungsthemenobjekt dargestellt wird, wird die Baumstruktur so lange nach oben verschoben, bis sich das nächstgelegene übergeordnete Verwaltungsthemenobjekt befindet. Das Sicherheitsprofil wird basierend auf dem Namen dieses Themenobjekts überprüft.

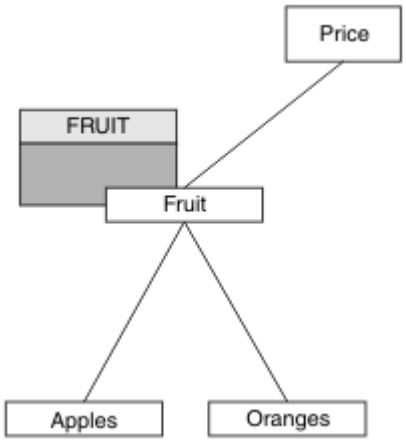


Abbildung 24. Beispiel für das Erteilen des Zugriffs auf ein Thema in einer Themenstruktur

Tabelle 87. Zugriffsvoraussetzungen für Beispielthemen und Themenobjekte			
	Thema	**Subskriptions-zugriff erforderlich**	**Themenobjekt**
Preis		Kein Benutzer	--
	Preis/>Obst	USER1	OBST
	Preis/Obst/Äpfel	USER1	
	Preis/Obst/Orangen	USER1	

In der vorherigen Task wurde USER1 Zugriff auf die Subskription des Themas `"Price/Fruit"` erteilt, indem ihm Zugriff auf das `hlq.SUBSCRIBE.FRUIT` -Profil unter z/OS und Subskriptionszugriff auf das `FRUIT` -Profil auf anderen Plattformen erteilt wurde. Dieses einzelne Profil erteilt auch USER1 Zugriff zum Subskribieren von `"Price/Fruit/Apples"`, `"Price/Fruit/Oranges"` und `"Price/Fruit/#"`.

Wenn USER1 versucht, das Thema `"Price/Fruit/Apples"` zu subskribieren, ist das Ergebnis erfolgreich.

Wenn USER2 versucht, das Thema `"Price/Fruit/Apples"` zu subskribieren, schlägt das Ergebnis mit einer `MQRC_NOT_AUTHORIZED` -Nachricht zusammen mit Folgendem fehl:

- Unter z/OS werden die folgenden Nachrichten in der Konsole angezeigt, die den vollständigen Sicherheitspfad über die Themenstruktur zeigen, die ausgeführt werden sollte:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- Auf anderen Plattformen ist das folgende Autorisierungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

Dabei ist Folgendes zu beachten:

- Die Nachrichten, die Sie in z/OS empfangen, sind mit denen identisch, die in der vorherigen Task empfangen wurden, da dieselben Themenobjekte und Profile den Zugriff steuern.
- Die Ereignisnachricht, die Sie auf anderen Plattformen erhalten, ist mit der in der vorherigen Task empfangenen Nachricht vergleichbar, die tatsächliche Themenzeichenfolge ist jedoch unterschiedlich.

Erteilen Sie einem anderen Benutzerzugriff, um nur das Thema tiefer in der Baumstruktur subskribieren zu können.

Dieses Thema ist die dritte in einer Liste mit Tasks, die Ihnen die Erteilung des Zugriffs auf die Subskription von Themen durch mehr als einen Benutzer erklärt.

Vorbereitende Schritte

In diesem Abschnitt wird die in „Einem Benutzer Zugriff gewähren, um ein Thema tiefer in der Baumstruktur zu subskribieren” auf Seite 527 beschriebene Konfiguration verwendet.

Informationen zu diesem Vorgang

In der vorherigen Task wurde USER2 der Zugriff auf das Thema `"Price/Fruit/Apples"`verweigert. In diesem Abschnitt erfahren Sie, wie Sie Zugriff auf dieses Thema erteilen können, aber nicht zu anderen Themen.

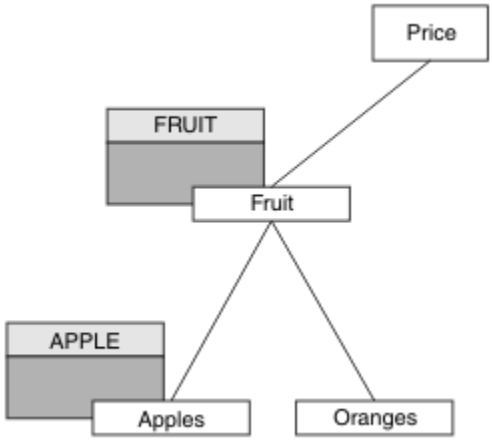


Abbildung 25. Zugriff auf bestimmte Themen in einer Themenstruktur erteilen

Tabelle 88. Zugriffsvoraussetzungen für Beispielthemen und Themenobjekte

Thema	Subskriptionszugriff erforderlich	Themenobjekt
Preis	Kein Benutzer	--
Preis/>Obst	USER1	OBST
Preis/Obst/Äpfel	BENUTZER1 und BENUTZER2	APFEL
Preis/Obst/Orangen	USER1	

Definieren Sie ein neues Themenobjekt wie folgt:

Vorgehensweise

1. Geben Sie den MQSC-Befehl `DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')`aus.

2. Gehen Sie wie folgt vor:

   - z/OS **z/OS** :

     In der vorherigen Task wurde USER1 Zugriff auf die Subskription des Themas `"Price/Fruit/Apples"` erteilt, indem dem Benutzer Zugriff auf das Profil `hlq.SUBSCRIBE.FRUIT` erteilt wurde.

     Dieses einzelne Profil hat auch USER1 -Zugriff auf die Subskription von `"Price/Fruit/Oranges"` `"Price/Fruit/#"` . Dieser Zugriff bleibt auch nach dem Hinzufügen des neuen Themenobjekts und der zugehörigen Profile erhalten.

     Erteilen Sie Zugriff auf USER2 , um das Thema `"Price/Fruit/Apples"` zu subskribieren, indem Sie dem Benutzer Zugriff auf das `hlq.SUBSCRIBE.APPLE` -Profil erteilen. Führen Sie dazu die folgenden RACF-Befehle aus:

     ```
     RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
     PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
     ```

   - Andere Plattformen:

     In der vorherigen Task wurde USER1 Zugriff zum Subskribieren des Themas `"Price/Fruit/Apples"` erteilt, indem dem Benutzer Subskriptionszugriff auf das FRUIT -Profil erteilt wurde.

     Dieses einzelne Profil hat auch USER1 -Zugriff auf die Subskription von `"Price/Fruit/Oranges"` und `"Price/Fruit/#"`, und dieser Zugriff bleibt auch nach dem Hinzufügen des neuen Themenobjekts und der zugehörigen Profile erhalten.

     Erteilen Sie den Zugriff auf USER2 , um das Thema `"Price/Fruit/Apples"` zu subskribieren, indem Sie dem Benutzer den Subskriptionszugriff auf das APPLE -Profil erteilen. Führen Sie dazu den Berechtigungsbefehl für die Plattform aus:

     ALW **AIX, Linux, and Windows-Systeme**

     ```
     setmqaut -t topic -n APPLE -p USER2 +sub
     ```

     IBM i **IBM i**

     ```
     GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*SUB)
     ```

Ergebnisse

Wenn USER1 unter z/OSversucht, das Thema `"Price/Fruit/Apples"` zu subskribieren, schlägt die erste Sicherheitsprüfung im `hlq.SUBSCRIBE.APPLE` -Profil fehl, aber beim Verschieben der Baumstruktur nach oben ermöglicht das `hlq.SUBSCRIBE.FRUIT` -Profil das Subskribieren von USER1 , sodass die Subskription erfolgreich ist und kein Rückkehrcode an den MQSUB-Aufruf gesendet wird. Für die erste Prüfung wird allerdings die RACF-Nachricht `ICH` generiert:

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

Wenn USER2 versucht, das Thema `"Price/Fruit/Apples"` zu subskribieren, ist das Ergebnis erfolgreich, da die Sicherheitsprüfung das erste Profil durchläuft.

Wenn USER2 versucht, das Thema `"Price/Fruit/Oranges"` zu subskribieren, schlägt das Ergebnis mit einer `MQRC_NOT_AUTHORIZED` -Nachricht zusammen mit Folgendem fehl:

- z/OS Unter z/OS werden die folgenden Nachrichten in der Konsole angezeigt, die den vollständigen Sicherheitspfad über die Themenstruktur zeigen, die ausgeführt werden sollte:

  ```
  ICH408I USER(USER2   ) ...
  ```

```
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW Auf AIX, Linux, and Windows-Plattformen das folgende Berechtigungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- IBM i Unter IBMi das folgende Berechtigungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

Der Nachteil dieser Konfiguration ist, dass Sie in z/OS zusätzliche ICH-Nachrichten an der Konsole erhalten. Sie können dies vermeiden, wenn Sie die Themenstruktur auf eine andere Weise sichern.

Zugriffssteuerung ändern, um zusätzliche Nachrichten zu vermeiden

Dieses Thema ist das vierte Thema in einer Liste mit Tasks, in denen der Zugriff zum Suskribieren von Themen durch mehrere Benutzer und das Vermeiden zusätzlicher RACF ICH408I-Nachrichten unter z/OS beschrieben wird.

Vorbereitende Schritte

In diesem Abschnitt wird die in „Erteilen Sie einem anderen Benutzerzugriff, um nur das Thema tiefer in der Baumstruktur subskribieren zu können.” auf Seite 529 beschriebene Konfiguration verbessert, so dass Sie zusätzliche Fehlernachrichten vermeiden können.

Informationen zu diesem Vorgang

In diesem Abschnitt erfahren Sie, wie Sie den Zugriff auf Themen vertiefen, die in der Baumstruktur enthalten sind, und wie Sie den Zugriff auf das Thema unten in der Baumstruktur entfernen können, wenn es kein Benutzer benötigt.

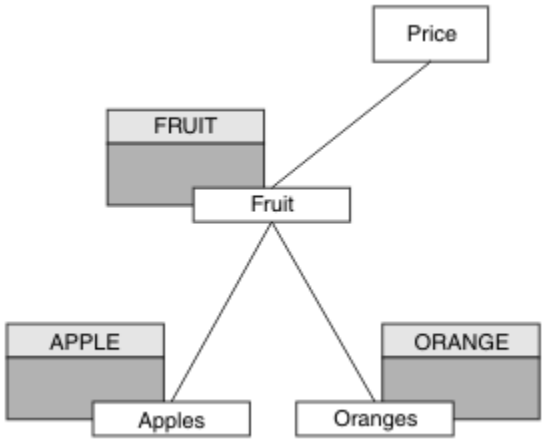


Abbildung 26. Beispiel für das Erteilen der Zugriffssteuerung, um zusätzliche Nachrichten zu vermeiden.

Definieren Sie ein neues Themenobjekt wie folgt:

Vorgehensweise

1. Geben Sie den MQSC-Befehl `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')` aus.
2. Gehen Sie wie folgt vor:

   - z/OS **z/OS** :

     Definieren Sie ein neues Profil und fügen Sie Zugriff auf dieses Profil und die vorhandenen Profile hinzu. Führen Sie dazu die folgenden RACF-Befehle aus:

     ```
     RDEFINE MXTOPIC hlq.SUBSCRIBE.ORANGE UACC(NONE)
     PERMIT hlq.SUBSCRIBE.ORANGE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
     PERMIT hlq.SUBSCRIBE.APPLE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
     ```

   - Andere Plattformen:

     Richten Sie den entsprechenden Zugriff mithilfe der Berechtigungsbefehle für die Plattform ein:

     ALW **AIX, Linux, and Windows-Systeme**

     ```
     setmqaut -t topic -n ORANGE -p USER1 +sub
     setmqaut -t topic -n APPLE -p USER1 +sub
     ```

     IBM i **IBM i**

     ```
     GRTMQAUT OBJ(ORANGE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
     GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
     ```

Ergebnisse

Wenn USER1 unter z/OSversucht, das Thema `"Price/Fruit/Apples"` zu subskribieren, ist die erste Sicherheitsprüfung für das Profil `hlq.SUBSCRIBE.APPLE` erfolgreich.

Wenn USER2 versucht, das Thema `"Price/Fruit/Apples"` zu subskribieren, ist das Ergebnis erfolgreich, weil die Sicherheitsprüfung das erste Profil durchläuft.

Wenn USER2 versucht, das Thema `"Price/Fruit/Oranges"` zu subskribieren, schlägt das Ergebnis mit einer `MQRC_NOT_AUTHORIZED` -Nachricht zusammen mit Folgendem fehl:

- z/OS Unter z/OS werden die folgenden Nachrichten in der Konsole angezeigt, die den vollständigen Sicherheitspfad über die Themenstruktur zeigen, die ausgeführt werden sollte:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.ORANGE ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW Auf anderen Plattformen ist das folgende Autorisierungsereignis:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

- IBM i Unter IBMi das folgende Berechtigungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

Einem Benutzer den Zugriff auf die Veröffentlichung in einem Thema gewähren

Dieses Thema ist die erste in einer Liste mit Tasks, in denen Sie erfahren, wie Sie den Zugriff auf Veröffentlichungsthemen von mehr als einem Benutzer erteilen können.

Informationen zu diesem Vorgang

Bei dieser Task wird davon ausgegangen, dass keine Verwaltungsthemenobjekte auf der rechten Seite der Themenstruktur vorhanden sind und dass keine Profile für die Veröffentlichung definiert wurden. Die Voraussetzung dafür ist, dass Publisher nur die Themenzeichenfolge verwenden.

Eine Anwendung kann in einem Thema veröffentlichen, indem sie ein Themenobjekt oder eine Themenzeichenfolge oder eine Kombination aus beiden bereitstellt. Whichever Art und Weise, wie die Anwendung ausgewählt wird, ist die Veröffentlichung an einem bestimmten Punkt in der Themenstruktur zu veröffentlichen. Wenn dieser Punkt in der Themenstruktur durch ein Verwaltungsthemenobjekt dargestellt wird, wird ein Sicherheitsprofil basierend auf dem Namen dieses Themenobjekts überprüft. Beispiel:

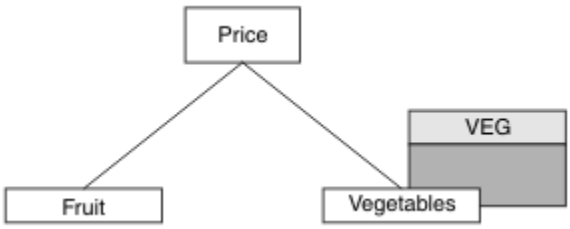


Abbildung 27. Publizierungszugriff auf ein Thema erteilen

Tabelle 89. Beispiel für Veröffentlichungszugriffs-Anforderungen

Thema	Publizier-Zugriff	Themenobjekt
Preis	Kein Benutzer	--
Preis/Gemüse	USER1	VEG

Definieren Sie ein neues Themenobjekt wie folgt:

Vorgehensweise

1. Geben Sie den MQSC-Befehl `DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')` aus.
2. Gehen Sie wie folgt vor:
   - z/OS **z/OS** :

     Erteilen Sie dem Benutzer Zugriff auf USER1 für die Veröffentlichung im Thema `"Price/Vegetables"` , indem Sie dem Benutzer Zugriff auf das `hlq.PUBLISH.VEG` -Profil erteilen. Führen Sie dazu die folgenden RACF-Befehle aus:

     ```
     RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
     PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
     ```

   - Andere Plattformen:

Erteilen Sie dem Benutzer Zugriff auf USER1 für die Veröffentlichung im Thema `"Price/Vegetables"` , indem Sie dem Benutzer Zugriff auf das VEG -Profil erteilen. Führen Sie dazu den Berechtigungsbefehl für die Plattform aus:

AIX, Linux, and Windows-Systeme

```
setmqaut -t topic -n VEG -p USER1 +pub
```

IBM i

```
GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
```

Ergebnisse

Wenn USER1 versucht, Nachrichten zum Thema `"Price/Vegetables"` zu veröffentlichen, ist das Ergebnis erfolgreich, d. h., der MQOPEN-Aufruf ist erfolgreich.

Wenn USER2 versucht, Nachrichten im Thema `"Price/Vegetables"` zu veröffentlichen, schlägt der MQOPEN-Aufruf mit einer `MQRC_NOT_AUTHORIZED` -Nachricht fehl, zusammen mit:

- Unter z/OS werden die folgenden Nachrichten in der Konsole angezeigt, die den vollständigen Sicherheitspfad über die Themenstruktur zeigen, die ausgeführt werden sollte:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Auf anderen Plattformen ist das folgende Autorisierungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

- Unter IBMi das folgende Berechtigungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

Beachten Sie, dass es sich hier um ein Beispiel für das, was Sie sehen, nicht um alle Felder handelt.

Einem Benutzer Zugriff gewähren, um die Veröffentlichung in einem Thema innerhalb der Baumstruktur zu veröffentlichen

Dieses Thema ist die zweite in einer Taskliste, in der Sie erfahren, wie Sie den Zugriff auf die Veröffentlichung von Themen durch mehr als einen Benutzer erteilen.

Vorbereitende Schritte

In diesem Abschnitt wird die in „Einem Benutzer den Zugriff auf die Veröffentlichung in einem Thema gewähren” auf Seite 533 beschriebene Konfiguration verwendet.

Informationen zu diesem Vorgang

Wenn der Punkt in der Themenstruktur, in dem die Anwendung veröffentlicht wird, nicht durch ein Verwaltungsthemenobjekt dargestellt wird, wird die Baumstruktur so lange nach oben verschoben, bis sich das nächstgelegene übergeordnete Verwaltungsthemenobjekt befindet. Das Sicherheitsprofil wird basierend auf dem Namen dieses Themenobjekts überprüft.

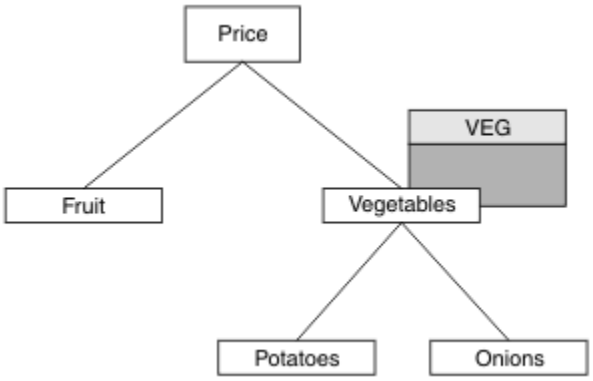


Abbildung 28. Publizierungszugriff auf ein Thema in einer Themenstruktur erteilen

Tabelle 90. Beispiel für Veröffentlichungszugriffs-Anforderungen

Thema	Subskriptionszugriff erforderlich	Themenobjekt
Preis	Kein Benutzer	--
Preis/Gemüse	USER1	VEG
Preis/Gemüse/Kartoffeln	USER1	
Preis/Gemüse/Zwiebeln	USER1	

In der vorherigen Task wurde USER1 Zugriff für die Veröffentlichung des Themas `"Price/Vegetables/Potatoes"` erteilt, indem Zugriff auf das Profil `hlq.PUBLISH.VEG` unter z/OS oder Veröffentlichungszugriff auf das Profil VEG auf anderen Plattformen erteilt wurde. Dieses einzelne Profil gewährt auch USER1 Zugriff zum Veröffentlichen unter `"Price/Vegetables/Onions"`.

Wenn USER1 versucht, beim Thema `"Price/Vegetables/Potatoes"` zu veröffentlichen, ist das Ergebnis erfolgreich, d. h., der MQOPEN-Aufruf ist erfolgreich.

Wenn USER2 versucht, das Thema `"Price/Vegetables/Potatoes"` zu subskribieren, ist das Ergebnis ein Fehler. Das heißt, der MQOPEN-Aufruf schlägt mit einer `MQRC_NOT_AUTHORIZED` -Nachricht fehl, zusammen mit:

- Unter z/OS werden die folgenden Nachrichten in der Konsole angezeigt, die den vollständigen Sicherheitspfad über die Themenstruktur zeigen, die ausgeführt werden sollte:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Auf anderen Plattformen ist das folgende Autorisierungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
```

```
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

Dabei ist Folgendes zu beachten:

- Die Nachrichten, die Sie in z/OS empfangen, sind mit denen identisch, die in der vorherigen Task empfangen wurden, da dieselben Themenobjekte und Profile den Zugriff steuern.
- Die Ereignisnachricht, die Sie auf anderen Plattformen erhalten, ist mit der in der vorherigen Task empfangenen Nachricht vergleichbar, die tatsächliche Themenzeichenfolge ist jedoch unterschiedlich.

Zugriff für Publish/Subscribe erteilen

Dieses Thema ist der letzte in einer Taskliste, in der Sie erfahren, wie Sie Zugriff zum Veröffentlichen und Subskribieren von Themen durch mehr als einen Benutzer erteilen.

Vorbereitende Schritte

In diesem Abschnitt wird die in „Einem Benutzer Zugriff gewähren, um die Veröffentlichung in einem Thema innerhalb der Baumstruktur zu veröffentlichen" auf Seite 534 beschriebene Konfiguration verwendet.

Informationen zu diesem Vorgang

In einer früheren Aufgabe wurde USER1 Zugriff zum Subskribieren des Themas `"Price/Fruit"` erhalten. In diesem Abschnitt erfahren Sie, wie Sie diesem Benutzer den Zugriff auf die Veröffentlichung zu diesem Thema gewähren.

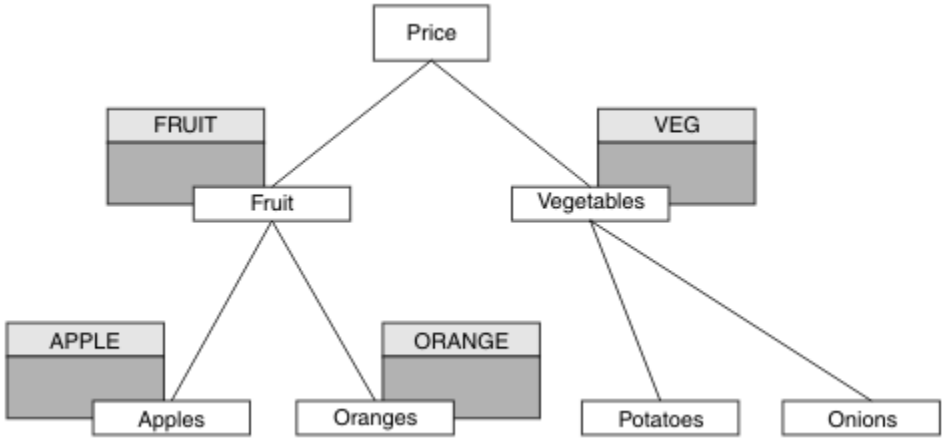


Abbildung 29. Zugriff für Veröffentlichung und Subskribierung erteilen

Tabelle 91. Beispiel für Veröffentlichungs-und Subskribierungszugriffanforderungen

Thema	Subskriptionszugriff erforderlich	Publizier-Zugriff	Themenobjekt
Preis	Kein Benutzer	Kein Benutzer	--
Preis/>Obst	USER1	USER1	OBST
Preis/Obst/Äpfel	BENUTZER1 und BENUTZER2		APFEL
Preis/Obst/Orangen	USER1		ORANGE

Vorgehensweise

Gehen Sie wie folgt vor:

- z/OS **z/OS** :

  In einer früheren Task wurde USER1 Zugriff auf die Subskription des Themas "Price/Fruit" erteilt, indem dem Benutzer Zugriff auf das Profil hlq.SUBSCRIBE.FRUIT erteilt wurde.

  Erteilen Sie für die Veröffentlichung im Thema "Price/Fruit" den Zugriff auf USER1 für das Profil hlq.PUBLISH.FRUIT . Führen Sie dazu die folgenden RACF-Befehle aus:

```
RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
```

- Andere Plattformen:

  Erteilen Sie dem Benutzer Veröffentlichungszugriff auf das FRUIT -Profil, um USER1 die Veröffentlichung zum Thema "Price/Fruit" zu ermöglichen. Führen Sie dazu den Berechtigungsbefehl für die Plattform aus:

  ALW **AIX, Linux, and Windows-Systeme**

```
setmqaut -t topic -n FRUIT -p USER1 +pub
```

  IBM i **IBM i**

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
```

Ergebnisse

Wenn USER1 unter z/OSversucht, im Thema "Price/Fruit" zu veröffentlichen, besteht die Sicherheitsüberprüfung im MQOPEN-Aufruf.

Wenn USER2 versucht, beim Thema "Price/Fruit" zu veröffentlichen, schlägt das Ergebnis mit einer MQRC_NOT_AUTHORIZED -Nachricht zusammen mit Folgendem fehl:

- z/OS Unter z/OS werden die folgenden Nachrichten in der Konsole angezeigt, die den vollständigen Sicherheitspfad über die Themenstruktur zeigen, die ausgeführt werden sollte:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ALW Auf AIX, Linux, and Windows-Plattformen das folgende Berechtigungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i Auf IBM i das folgende Berechtigungsereignis:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
```

```
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

Nach der vollständigen Gruppe dieser Tasks erteilt USER1 und USER2 die folgenden Zugriffsberechtigungen für Publish/Subscribe für die aufgelisteten Themen:

Tabelle 92. Vollständige Liste der Zugriffsberechtigungen, die sich aus Sicherheitsbeispielen ergeben

Thema	Subskriptionszugriff erforderlich	Publizier-Zugriff	Themenobjekt
Preis	Kein Benutzer	Kein Benutzer	--
Preis/>Obst	USER1	USER1	OBST
Preis/Obst/Äpfel	BENUTZER1 und BENUTZER2		APFEL
Preis/Obst/Orangen	USER1		ORANGE
Preis/Gemüse		USER1	VEG
Preis/Gemüse/Kartoffeln			
Preis/Gemüse/Zwiebeln			

Wenn Sie unterschiedliche Anforderungen für den Sicherheitszugriff auf unterschiedlichen Ebenen in der Themenstruktur haben, stellt eine sorgfältige Planung sicher, dass Sie keine überzähligen Sicherheitswarnungen im z/OS-Konsolenprotokoll erhalten. Durch die Einstellung der Sicherheit auf der richtigen Ebene innerhalb des Baums werden irreführende Sicherheitsnachrichten vermieden.

Subskriptionssicherheit

MQSO_ALTERNATE_USER_AUTHORITY

Das Feld AlternateUserId enthält eine Benutzer-ID zur Überprüfung dieses MQSUB-Aufrufs. Der Aufruf kann nur dann erfolgreich sein, wenn diese Alternative-Benutzer-ID berechtigt ist, das Thema mit den angegebenen Zugriffsoptionen zu subskribieren, unabhängig davon, ob die Benutzer-ID, unter der die Anwendung ausgeführt wird, berechtigt ist, dies zu tun.

MQSO_SET_IDENTITY_CONTEXT

Die Subskription ist die Verwendung der in den Feldern 'PubAccountingToken' und 'PubApplIdentityData' bereitgestellten Daten zur Accountkennung und zur Anwendungsidentität.

Wenn diese Option angegeben wird, wird dieselbe Berechtigungsprüfung ausgeführt, als wäre der Zugriff auf die Zielwarteschlange über einen MQOPEN-Aufruf mit MQOO_SET_IDENTITY_CONTEXT erfolgt. Dies gilt nicht für den Fall, dass die Option MQSO_MANAGED ebenfalls verwendet wird. In diesem Fall erfolgt keine Berechtigungsprüfung in der Zielwarteschlange.

Wenn diese Option nicht angegeben wird, sind den Veröffentlichungen, die an diesen Subskribenten gesendet werden, folgende Standardkontextinformationen zugeordnet:

Tabelle 93. Standardinformationen zu Veröffentlichungskontexten	
Feld im MQMD	**Verwendeter Wert**
UserIdentifier	Die Benutzer-ID, die der Subskription zugeordnet ist (siehe SUBUSER-Feld in DISPLAY SBSTATUS) zum Zeitpunkt der Veröffentlichung der Veröffentlichung.
AccountingToken	Wird, wenn möglich, durch die Umgebung bestimmt; wird andernfalls auf MQACT_NONE gesetzt.
ApplIdentityData	Wird auf Leerzeichen gesetzt.

Diese Option ist nur mit MQSO_CREATE und MQSO_ALTER gültig. Bei Verwendung mit MQSO_RESUME werden die Felder "PubAccountingToken" und "PubApplIdentityData" ignoriert, so dass diese Option keine Auswirkungen hat.

Wird eine Subskription, von der zuvor identitätsbezogene Kontextinformationen bereitgestellt wurden, ohne diese Option geändert, werden für die geänderte Subskription standardmäßige Kontextinformationen generiert.

Wenn eine Subskription, die zulässt, dass verschiedene Benutzer-IDs sie mit der Option MQSO_ANY_USERID verwenden, von einer anderen Benutzer-ID fortgesetzt wird, wird ein Standardidentitätskontext für die neue Benutzer-ID generiert, die jetzt Eigner der Subskription ist. Alle nachfolgenden Veröffentlichungen werden mit dem neuen Identitätskontext bereitgestellt.

AlternateSecurityId

Dies ist eine Sicherheits-ID, die mit der AlternateUserId an den Berechtigungsservice übergeben wird, damit entsprechende Berechtigungsprüfungen ausgeführt werden können. AlternateSecurityId wird nur verwendet, wenn MQSO_ALTERNATE_USER_AUTHORITY angegeben ist und das Feld AlternateUserId nicht bis zum ersten Nullzeichen oder bis zum Ende des Felds vollständig leer ist.

MQSO_ANY_USERID, Subskriptionsoption

Wenn MQSO_ANY_USERID angegeben ist, ist die Identität des Subskribenten nicht auf eine einzelne Benutzer-ID eingeschränkt. Dadurch kann jeder Benutzer die Subskription ändern oder fortsetzen, sofern er über die entsprechende Berechtigung verfügt. Die Subskription kann jeweils nur einem einzelnen Benutzer gehören. Ein Versuch, die Verwendung einer Subskription wiederaufzunehmen, die derzeit von einer anderen Anwendung verwendet wird, führt dazu, dass der Aufruf mit MQRC_SUBSCRIPTION_IN_USE fehlschlägt.

Wenn Sie diese Option einer vorhandenen Subskription hinzufügen möchten, muss der MQSUB-Aufruf (mit MQSO_ALTER) von derselben Benutzer-ID stammen wie die ursprüngliche Subskription.

Wenn sich ein MQSUB-Aufruf auf eine vorhandene Subskription bezieht, für die MQSO_ANY_USERID festgelegt ist, und die Benutzer-ID von der ursprünglichen Subskription abweicht, ist der Aufruf nur erfolgreich, wenn die neue Benutzer-ID über die Berechtigung verfügt, das Thema zu abonnieren. Nach erfolgreichem Abschluss werden zukünftige Veröffentlichungen zu diesem Subskribenten in die Warteschlange des Subskribenten gestellt, wobei die neue Benutzer-ID in der Veröffentlichung festgelegt ist.

MQSO_FIXED_USERID

Wenn MQSO_FIXED_USERID angegeben ist, kann die Subskription nur von einer einzigen Benutzer-ID geändert oder wieder aufgenommen werden, die Eigner ist. Diese Benutzer-ID ist die letzte Benutzer-ID, mit der die Subskription geändert wird, die diese Option definiert, wodurch die Option MQSO_ANY_USERID entfernt wird, oder wenn keine Änderungen stattgefunden haben, ist dies die Benutzer-ID, die die Subskription erstellt hat.

Wenn ein MQSUB-Verb auf eine vorhandene Subskription mit der Gruppe MQSO_ANY_USERID verweist und die Subskription (mit MQSO_ALTER) ändert, um die Option MQSO_FIXED_USERID zu verwenden, wird die Benutzer-ID der Subskription jetzt an dieser neuen Benutzer-ID festgelegt. Der Aufruf ist nur erfolgreich, wenn die neue Benutzer-ID befugt ist, das Thema zu abonnieren.

Wenn eine andere Benutzer-ID als die, die als Eigentümer einer Subskription für die Wiederaufnahme oder Änderung einer MQSO_FIXED_USERID-Subskription aufgezeichnet wurde, fehlschlägt, schlägt der Aufruf mit MQRC_IDENTITY_MISMATCH fehl. Die Benutzer-ID, die Eigner einer Subskription ist, kann mit dem Befehl DISPLAY SBSTATUS angezeigt werden.

Wenn weder MQSO_ANY_USERID noch MQSO_FIXED_USERID angegeben ist, wird der Standardwert MQSO_FIXED_USERID verwendet.

Publish/Subscribe-Sicherheit zwischen Warteschlangenmanagern

Publish/Subscribe-interne Nachrichten, wie z. B. Proxy-Subskriptionen und Veröffentlichungen, werden mithilfe der normalen Kanalsicherheitsregeln in Warteschlangen für Publish/Subscribe-Systeme gestellt. In den Informationen und Diagrammen in diesem Thema werden die verschiedenen Prozesse und Benutzer-IDs hervorgehoben, die an der Zustellung dieser Nachrichten beteiligt sind.

Lokale Zugriffssteuerung

Der Zugriff auf Themen für Veröffentlichungen und Subskriptionen richtet sich nach lokalen Sicherheitsdefinitionen und -regeln, die in der Publish/Subscribe-Sicherheit beschrieben sind. Unter z/OS ist kein lokales Themenobjekt erforderlich, um die Zugriffssteuerung zu erstellen. Für die Zugriffssteuerung auf anderen Plattformen ist kein lokales Thema erforderlich. Administratoren können die Zugriffssteuerung auf Clusterthemenobjekte anwenden, unabhängig davon, ob sie noch im Cluster vorhanden sind.

Systemadministratoren sind für die Zugriffssteuerung auf ihrem lokalen System verantwortlich. Sie müssen den Administratoren anderer Mitglieder der Hierarchie oder Cluster-Brokerverbünden vertrauen, die für ihre Zugriffssteuerungsrichtlinie verantwortlich sind. Da die Zugriffssteuerung für jede einzelne Maschine definiert ist, ist es wahrscheinlich, dass sie belastet wird, wenn eine Feinsteuerungskontrolle erforderlich ist. Es kann nicht erforderlich sein, eine Zugriffssteuerung zu erzwingen, oder die Zugriffssteuerung kann auf übergeordneten Objekten in der Themenstruktur definiert werden. Die Zugriffssteuerung auf Feinebene kann für jede Unterteilung des Topic-Namespace definiert werden.

Proxy-Subskription erstellen

Das Vertrauen für eine Organisation, die ihren Warteschlangenmanager mit Ihrem Warteschlangenmanager verbindet, wird durch die normalen Kanalauthentifizierungsmittel bestätigt. Wenn diese vertrauenswürdige Organisation auch verteilte Publish/Subscribe-Verfahren ausführen darf, wird eine Berechtigungs-Prüfung durchgeführt. Die Prüfung wird durchgeführt, wenn der Kanal eine Nachricht in eine verteilte Publish/Subscribe-Warteschlange einreiht. Beispiel: Eine Nachricht wird in die Warteschlange `SYSTEM.INTER.QMGR.CONTROL` gestellt. Die Benutzer-ID für die Warteschlangenberechtigungsüberprüfung hängt von den `PUTAUT` -Werten des empfangenden Kanals ab. Beispiel: Die Benutzer-ID des Kanals `MCAUSER` , der Nachrichtenkontext, abhängig von dem Wert und der Plattform. Weitere Informationen zur Kanalsicherheit finden Sie unter Kanalsicherheit .

Proxy-Subskriptionen werden mit der Benutzer-ID des verteilten Publish/Subscribe-Agenten auf dem fernen WS-Manager erstellt. Beispiel: `QM2` in Abbildung 30 auf Seite 541. Der Benutzer erhält dann problemlos Zugriff auf lokale Themenobjektprofile, da diese Benutzer-ID im System definiert ist und es daher keine Domänenkonflikte gibt.

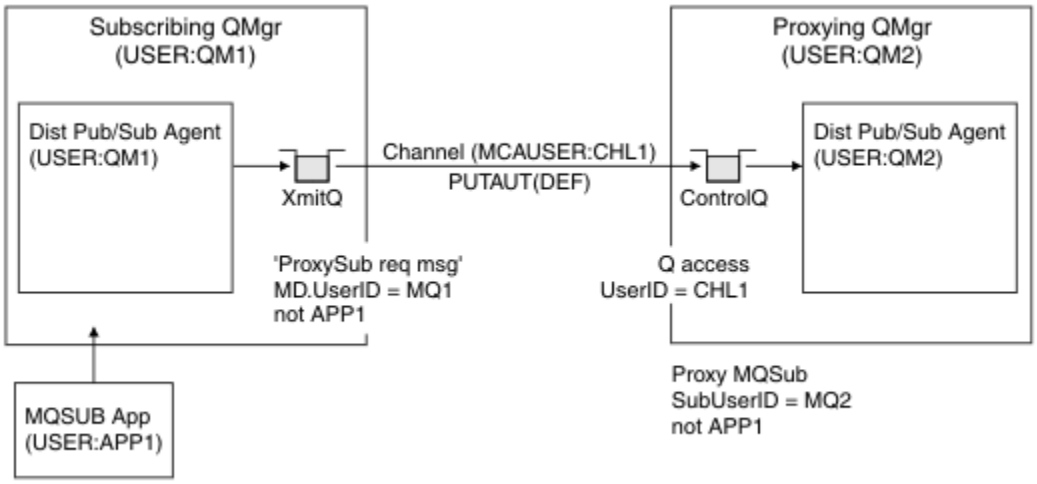


Abbildung 30. Proxy-Abonnementsicherheit, Subskription vornehmen

Zurücksenden von fernen Veröffentlichungen

Wenn eine Veröffentlichung auf dem Veröffentlichungswarteschlangenmanager erstellt wird, wird eine Kopie der Veröffentlichung für jede Proxy-Subskription erstellt. Der Kontext der kopierten Veröffentlichung enthält den Kontext der Benutzer-ID, die die Subskription erstellt hat; QM2 in Abbildung 31 auf Seite 541. Die Proxy-Subskription wird mit einer Zielwarteschlange erstellt, bei der es sich um eine ferne Warteschlange handelt, so dass die Veröffentlichungsnachricht in eine Übertragungswarteschlange aufgelöst wird.

Das Vertrauen in eine Organisation, die ihren Warteschlangenmanager QM2 mit einem anderen Warteschlangenmanager verbindet,QM1 wird durch normale Kanalauthentifizierungsmittel bestätigt. Wenn diese vertrauenswürdige Organisation dann zur/zum verteilten Veröffentlichung/Abonnement berechtigt wird, wird eine Berechtigungsprüfung durchgeführt, wenn der Kanal die Veröffentlichungsnachricht in die WarteschlangeSYSTEM.INTER.QMGR.PUBS für die/das verteilte Veröffentlichung/Abonnement stellt. Die Benutzer-ID für die Warteschlangenberechtigungsüberprüfung hängt vom Wert PUTAUT des empfangenden Kanals ab (z. B. die Benutzer-ID des Kanals, MCAUSER , Nachrichtenkontext und andere, abhängig von Wert und Plattform). Weitere Informationen zur Kanalsicherheit finden Sie unter Kanalsicherheit .

Wenn die Veröffentlichungsnachricht den Subskribentenwarteschlangenmanager erreicht, wird unter der Berechtigung dieses Warteschlangenmanagers ein weiterer MQPUT-Aufruf ausgeführt, und der Kontext mit der Nachricht wird durch den Kontext jedes lokalen Subskribenten ersetzt, da sie jeweils die Nachricht erhalten.

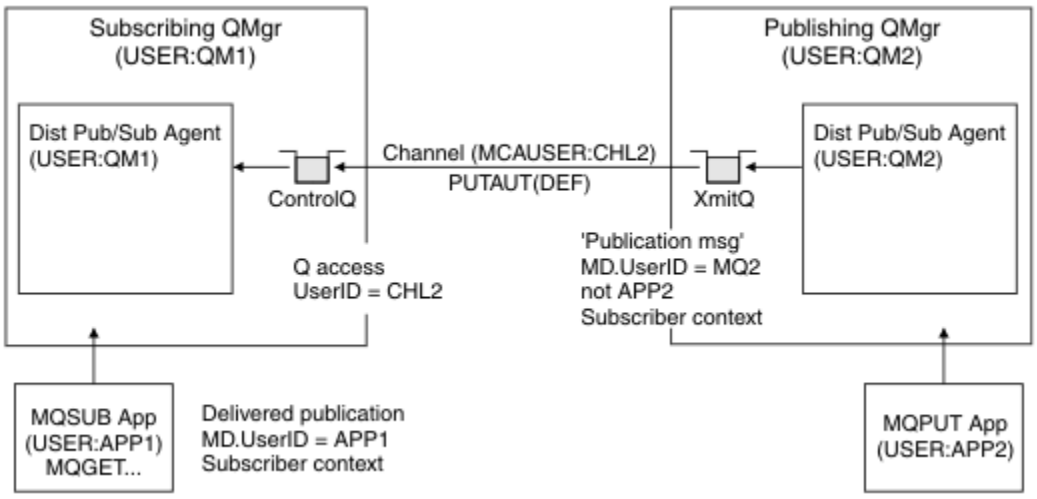


Abbildung 31. Proxy-Abonnementsicherheit, Weiterleitungsveröffentlich

In einem System, in dem wenig auf die Sicherheit geachtet wurde, laufen die verteilten Veröffentlichungs-/Abonnementprozesse wahrscheinlich unter einer Benutzer-ID in dermqm Gruppe, der MCAUSER-Parameter eines Kanals ist leer (Standard), und die Nachrichten werden nach Bedarf an die verschiedenen Systemwarteschlangen geliefert. Das ungesicherte System macht es einfach, einen Proof-of-Concept einzurichten, um verteiltes Publish/Subscribe zu demonstrieren.

Auf einem System, auf dem die Sicherheit ernsthaft in Betracht gezogen wird, unterliegen diese internen Nachrichten denselben Sicherheitssteuerungen wie jede Nachricht, die über den Kanal gesendet wird.

Wenn der Kanal mit einem nicht leeren MCAUSER und einem PUTAUT-Wert eingerichtet wird, der angibt, dass der MCAUSER überprüft werden muss, dann muss dem betreffenden MCAUSER der Zugang zu denSYSTEM.INTER.QMGR.* Warteschlangen gewährt werden. Wenn mehrere remote Warteschlangenmanager mit Kanälen unter unterschiedlichen MCAUSER -IDs vorhanden sind, müssen alle diese Benutzer-IDs Zugriff auf die SYSTEM.INTER.QMGR.* -Warteschlangen erhalten. Kanäle, die unter verschiedenen MCAUSER -IDs ausgeführt werden, können z. B. auftreten, wenn mehrere hierarchische Verbindungen auf einem einzigen Warteschlangenmanager konfiguriert sind.

Wenn der Kanal mit einem PUTAUT -Wert eingerichtet wird, der angibt, dass der Kontext der Nachricht verwendet wird, wird der Zugriff auf die SYSTEM.INTER.QMGR.* -Warteschlangen basierend auf der Benutzer-ID innerhalb der internen Nachricht überprüft. Da alle diese Nachrichten mit der Benutzer-ID des verteilten Publish/Subscribe-Agenten aus dem Warteschlangenmanager, der die interne Nachricht sendet, oder einer Veröffentlichungsnachricht (siehe Abbildung 31 auf Seite 541) gestellt werden, ist es nicht zu groß, dass eine Gruppe von Benutzer-IDs Zugriff auf die verschiedenen Systemwarteschlangen (einen pro fernen Warteschlangenmanager) erteilt, wenn Sie Ihre verteilte Publish/Subscribe-Sicherheit auf diese Weise einrichten möchten. Es weist immer noch dieselben Probleme auf, die die Kanalkontextsicherheit immer hat; die der verschiedenen Benutzer-ID-Domänen und die Tatsache, dass die Benutzer-ID in der Nachricht möglicherweise nicht auf dem empfangenden System definiert ist. Es ist jedoch eine absolut akzeptable Möglichkeit, bei Bedarf auszuführen.

z/OS Sicherheit der Systemwarteschlange stellt eine Liste mit Warteschlangen und den Zugriff bereit, die für die sichere Einrichtung der verteilten Publish/Subscribe-Umgebung erforderlich ist. Wenn interne Nachrichten oder Veröffentlichungen aufgrund von Sicherheitsverletzungen nicht in die Warteschlange gestellt werden, schreibt der Kanal eine Nachricht in die normale Art und Weise in das Protokoll, und die Nachrichten können entsprechend der normalen Kanalfehlerverarbeitung an die Warteschlange für dead-letter gesendet werden.

Alle Messaging-Manager-Nachrichten für die Zwecke verteilter Publish/Subscribe-Nachrichten werden unter Verwendung der normalen Kanalsicherheit ausgeführt.

Informationen zum Einschränken von Veröffentlichungen und Proxy-Abonnements auf der Themenebene finden Sie im Abschnitt Veröffentlichungs/Abonnement-Sicherheit.

Standard-Benutzer-IDs mit einer WS-Manager-Hierarchie verwenden

Wenn Sie eine Hierarchie von WS-Managern haben, die auf verschiedenen Plattformen ausgeführt werden und die Standardbenutzer-IDs verwenden, beachten Sie, dass diese Standardbenutzer-IDs von den Plattformen abweichen und auf der Zielplattform möglicherweise nicht bekannt sind. Infolgedessen weist ein Warteschlangenmanager, der auf einer Plattform ausgeführt wird, Nachrichten zurück, die von Warteschlangenmanagern auf anderen Plattformen mit dem Ursachencode MQRC_NOT_AUTHORIZEDempfangen wurden.

Um Nachrichten zu vermeiden, die mindestens zurückgewiesen werden, müssen die folgenden Berechtigungen zu den Standardbenutzer-IDs hinzugefügt werden, die auf anderen Plattformen verwendet werden:

- Berechtigung *PUT *GET für die Warteschlange SYSTEM.BROKER. Warteschlangen
- Berechtigung *PUB *SUB für SYSTEM.BROKER. Themen
- Berechtigung *ADMCRT *ADMDLT *ADMCHG in der Warteschlange SYSTEM.BROKER.CONTROL.QUEUE.

Die Standardbenutzer-IDs mit einer Queue-Manager-Hierarchie sind wie folgt:

Plattform	Standardbenutzer-ID
Windows	`mqm`
Systeme mit AIX and Linux	`mqm`
IBM i	`QMQM`
z/OS	Die Benutzer-ID des Kanalinitiatoradressraums

Erstellen und gewähren Sie Zugriff auf die Benutzer-ID 'qmqm', wenn sie hierarchisch an einen Warteschlangenmanager in IBM i für Warteschlangenmanager auf z/OS, AIX, Linux, and Windows-Plattformen angehängt ist.

Bei Queue Managern auf IBM i- und z/OS-Plattformen erstellen und gewähren Sie Zugriff auf die Benutzer-ID 'mqm', wenn sie hierarchisch an einen Queue Manager unter AIX, Linux, and Windows angehängt sind.

Erstellen und erteilen Sie Benutzerzugriff auf die Benutzer-ID des Adressraums des z/OS-Kanalinitiators, wenn sie hierarchisch an einen Warteschlangenmanager in z/OS für Warteschlangenmanager unter Multiplatforms angehängt ist.

Bei Benutzer-IDs kann die Groß-/Kleinschreibung beachtet werden. Der ursprüngliche Warteschlangenmanager (sofern in Multiplatforms) erzwingt, dass die Benutzer-ID nur in Großbuchstaben angegeben wird. Der empfangende Warteschlangenmanager (sofern in AIX, Linux, and Windows) erzwingt, dass die Benutzer-ID nur in Kleinbuchstaben angegeben wird. Daher müssen alle Benutzer-IDs, die auf AIX and Linux-Systemen erstellt werden, in Kleinschreibung erstellt werden. Wenn ein Nachrichtenexit installiert wurde, wird die Benutzer-ID nicht in Großbuchstaben oder in Kleinbuchstaben umgesetzt. Es muss sorgfältig darauf geachtet werden, wie der Nachrichtenexit die Benutzer-ID verarbeitet.

Gehen Sie wie folgt vor, um mögliche Probleme bei der Konvertierung von Benutzer-IDs zu

- Stellen Sie auf Systemen mit AIX, Linux, and Windows sicher, dass die Benutzer-IDs in Kleinschreibung angegeben werden.
- Stellen Sie unter IBM i und z/OS sicher, dass die Benutzer-IDs in Großbuchstaben angegeben werden.

Sicherheit von IBM MQ Console und REST API

Die Sicherheit für IBM MQ Console und REST API wird durch Bearbeiten der Konfiguration des mqweb-Servers in der Datei `mqwebuser.xml` konfiguriert.

Informationen zu diesem Vorgang

Sie können Benutzeraktionen überwachen und die Verwendung der IBM MQ Console und der REST API prüfen, indem Sie die Protokolldateien des mqweb-Servers untersuchen.

Die Benutzer der IBM MQ Console und der REST API können mit folgenden Komponenten authentifiziert werden:

- Basisregistry
- LDAP-Registry
- Registry des lokalen Betriebssystems
- SAF unter z/OS
- Alle anderen Registry-Typen, die von WebSphere Liberty unterstützt werden

Rollen können IBM MQ Console- und REST API-Benutzern zugeordnet werden, um festzulegen, welche Zugriffsebene sie für IBM MQ-Objekte erhalten. Wenn Sie z. B. Messaging ausführen möchten, müssen Benutzer die Rolle `MQWebUser` zuordnen. Weitere Informationen zu den verfügbaren Rollen finden Sie unter „Rollen in der IBM MQ Console und der REST API” auf Seite 555.

Nachdem einem Benutzer eine Rolle zugeordnet wurde, gibt es eine Reihe von Methoden, die zur Authentifizierung des Benutzers verwendet werden können. Benutzer können sich mit einem Benutzernamen

und einem Kennwort oder über die Clientzertifikatsauthentifizierung an der IBM MQ Console anmelden. Mit dem REST API können Benutzer die HTTP-Basisauthentifizierung, die tokenbasierte Authentifizierung oder die Clientzertifikatsauthentifizierung verwenden.

Vorgehensweise

1. Definieren Sie die Benutzerregistry für die Authentifizierung von Benutzern und ordnen Sie jedem Benutzer oder jeder Gruppe eine Rolle zu, damit die Benutzer und Gruppen für die Verwendung der IBM MQ Console oder der REST API berechtigt sind. Weitere Informationen finden Sie in den folgenden Abschnitten:„Benutzer und Rollen konfigurieren" auf Seite 545
2. Wählen Sie aus, wie Benutzer der IBM MQ Console auf dem mqweb-Server authentifiziert werden sollen. Sie müssen nicht die gleiche Methode für alle Benutzer verwenden:
   - Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall gibt ein Benutzer im Anmeldebildschirm der IBM MQ Console eine Benutzer-ID und ein Kennwort ein. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Es ist keine weitere Konfiguration erforderlich, um diese Authentifizierungsoption zu verwenden, aber Sie können optional die Ablaufzeit für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt Ablaufintervall für LTPA-Token konfigurieren .
   - Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der IBM MQ Console, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden" auf Seite 561.
3. Wählen Sie aus, wie Benutzer der REST API auf dem mqweb-Server authentifiziert werden sollen. Sie müssen nicht die gleiche Methode für alle Benutzer verwenden:
   - Lassen Sie sich Benutzer authentifizieren, indem Sie die HTTP-Basisauthentifizierung verwenden In diesem Fall wird ein Benutzername und ein Kennwort codiert, aber nicht verschlüsselt, und mit jeder REST API-Anforderung gesendet, damit der Benutzer für diese Anforderung authentifiziert und berechtigt ist. Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Weitere Informationen finden Sie unter „HTTP-Basisauthentifizierung mit der REST API verwenden" auf Seite 565.
   - Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall stellt ein Benutzer der REST API `login`-Ressource mit der HTTP-POST-Methode eine Benutzer-ID und ein Kennwort bereit. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Weitere Informationen finden Sie unter „Tokenbasierte Authentifizierung mit der REST-API verwenden" auf Seite 566.

     Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Wenn Sie HTTP-Verbindungen aktiviert haben, können Sie jedoch ein LTPA-Token zulassen, das für eine HTTPS-Verbindung ausgegeben wird, die für eine HTTP-Verbindung verwendet werden soll. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.
   - Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der REST API, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden" auf Seite 561.
4. Optional: Konfigurieren Sie die Cross-Origin-Ressourcenfreigabe für die REST API.

   Standardmäßig sind im Web-Browser keine Scripts wie beispielsweise JavaScript für den Aufruf der REST API zulässig, wenn das Script nicht den gleichen Ursprung wie die REST API hat. Dies bedeutet, dass Kreuzursprungsanforderungen nicht aktiviert sind. Sie können Cross Origin Resource Sharing (CORS) konfigurieren, um Cross-Origin-Anforderungen von angegebenen URLs zu ermöglichen. Weitere Informationen finden Sie unter „CORS für die REST API konfigurieren" auf Seite 569.

5. Optional: Konfigurieren Sie die Validierung des Host-Headers für die IBM MQ Console und die REST API.

   Sie können die Validierung des Host-Headers konfigurieren und eine Zulassungsliste der Hostnamen und Ports erstellen, um sicherzustellen, dass nur Anforderungen mit bestimmten Host-Headern von der IBM MQ Console und der REST API verarbeitet werden. Weitere Informationen finden Sie unter „Validierung des Host-Headers für die IBM MQ Console und die REST API konfigurieren" auf Seite 570.

Benutzer und Rollen konfigurieren

Um die IBM MQ Console oder die REST API verwenden zu können, müssen Benutzer in einer Benutzerregistry authentifiziert sein, die für den mqweb-Server definiert ist.

Informationen zu diesem Vorgang

Authentifizierte Benutzer müssen Mitglied einer der Gruppen sein, die den Zugriff auf die Funktionen der IBM MQ Console und der REST API autorisieren. Die Benutzerregistry enthält standardmäßig keine Benutzer; diese müssen durch Bearbeiten der Datei `mqwebuser.xml` hinzugefügt werden.

Wenn Sie Benutzer und Gruppen konfigurieren, konfigurieren Sie zuerst eine Benutzerregistry, um Benutzer und Gruppen zu authentifizieren. Diese Benutzerregistry wird von der IBM MQ Console und REST APIgemeinsam genutzt. Sie können steuern, ob Benutzer und Gruppen Zugriff auf IBM MQ Console, REST API oder beide haben, wenn Sie Rollen für Ihre Benutzer und Gruppen konfigurieren.

Nachdem Sie die Benutzerregistry konfiguriert haben, konfigurieren Sie Rollen für die Benutzer und Gruppen, um ihnen die Berechtigung zu erteilen. Es sind mehrere Rollen verfügbar, einschließlich Rollen, die für die Verwendung von REST API für Managed File Transfer spezifisch sind. Jede Rolle gewährt eine andere Zugriffsebene. Weitere Informationen finden Sie unter „Rollen in der IBM MQ Console und der REST API" auf Seite 555.

Eine Reihe von XML-Musterdateien wird mit dem mqweb-Server bereitgestellt, um die Konfiguration von Benutzern und Gruppen zu vereinfachen. Benutzer mit Kenntnissen über die Konfiguration der Sicherheit in WebSphere Liberty (WLP) ziehen es möglicherweise vor, die Beispiele nicht zu verwenden. WLP stellt weitere Berechtigungsfunktionen bereit, die zusätzlich zu den hier dokumentierten Informationen bereitgestellt werden.

Prozedur

- Konfigurieren Sie Benutzer und Gruppen mit einer Basisregistry unter Verwendung der Datei `basic_registry.xml` .

  Mit den Benutzernamen und Kennwörtern in der Registry werden Benutzer der IBM MQ Console und der REST API authentifiziert und autorisiert.

  Informationen zum Konfigurieren einer Basisregistry unter Verwendung der Beispieldatei `basic_registry.xml` finden Sie in „Basisregistry für die IBM MQ Console und REST API konfigurieren" auf Seite 547.
- Konfigurieren Sie Benutzer und Gruppen mit einer LDAP-Registry mithilfe der `ldap_registry.xml` -Datei.

  Mit den Benutzernamen und Kennwörtern in der LDAP-Registry wird die Verwendung der IBM MQ Console und der REST API authentifiziert und autorisiert.

  Informationen zum Konfigurieren einer LDAP-Registry unter Verwendung der Beispieldatei `ldap_registry.xml` finden Sie in „LDAP-Registry für die IBM MQ Console und REST API konfigurieren" auf Seite 551.
- 

  Konfigurieren Sie Benutzer und Gruppen mit einer lokalen Betriebssystemregistry unter Verwendung der Datei `local_os_registry.xml` .

Mit den Benutzernamen und Kennwörtern in der Registry des Betriebssystems werden Benutzer der IBM MQ Console und der REST API authentifiziert und autorisiert.

Informationen zum Konfigurieren einer lokalen OS-Registry unter Verwendung der Beispieldatei `local_os_registry.xml` finden Sie in „Lokale OS-Registry für die IBM MQ Console und REST API konfigurieren" auf Seite 549.

- z/OS

  Konfigurieren Sie Benutzer und Gruppen mit der Systemberechtigungsfunktion (SAF = System Authorization Facility) unter z/OS mithilfe der Datei `zos_saf_registry.xml`.

  Mit RACF oder einem anderen Sicherheitsprodukt werden Profile verwendet, um Benutzern und Gruppen Zugriff auf Rollen zu erteilen. Mit den Benutzernamen und Kennwörtern in der RACF-Datenbank werden die Benutzer der IBM MQ Console und der REST API authentifiziert und autorisiert.

  Informationen zum Konfigurieren der SAF-Schnittstelle unter Verwendung der Beispieldatei `zos_saf_registry.xml` finden Sie in „SAF-Registry für die IBM MQ Console und REST API konfigurieren" auf Seite 553.
- Inaktivieren Sie die Sicherheit, einschließlich der Möglichkeit, über HTTPS auf die IBM MQ Console oder REST API zuzugreifen, indem Sie die Datei `no_security.xml` verwenden.

Nächste Schritte

Wählen Sie aus, wie Benutzer sich authenti

IBM MQ Console-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall gibt ein Benutzer im Anmeldebildschirm der IBM MQ Console eine Benutzer-ID und ein Kennwort ein. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Es ist keine weitere Konfiguration erforderlich, um diese Authentifizierungsoption zu verwenden, aber Sie können optional das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt Ablaufintervall für LTPA-Token konfigurieren .
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der IBM MQ Console, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden" auf Seite 561.

REST API-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die HTTP-Basisauthentifizierung verwenden In diesem Fall wird ein Benutzername und ein Kennwort codiert, aber nicht verschlüsselt, und mit jeder REST API-Anforderung gesendet, damit der Benutzer für diese Anforderung authentifiziert und berechtigt ist. Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Weitere Informationen finden Sie unter „HTTP-Basisauthentifizierung mit der REST API verwenden" auf Seite 565.
- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall stellt ein Benutzer der REST API `login`-Ressource mit der HTTP-POST-Methode eine Benutzer-ID und ein Kennwort bereit. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Weitere Informationen finden Sie unter „Tokenbasierte Authentifizierung mit der REST-API verwenden" auf Seite 566. Sie können das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der REST API, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden" auf Seite 561.

Basisregistry für die IBM MQ Console und REST API konfigurieren

Sie können eine Basisregistry in der `mqwebuser.xml` -Datei konfigurieren. Die Benutzernamen, Kennwörter und Rollen in der XML-Datei werden verwendet, um die Benutzer von IBM MQ Console und REST API zu authentifizieren und zu berechtigen.

Vorbereitende Schritte

- Wenn Sie Benutzer in der Basisregistry konfigurieren, müssen Sie jedem Benutzer eine Rolle zuordnen. Jede Rolle stellt verschiedene Berechtigungsebenen für den Zugriff auf die IBM MQ Console und die REST API bereit und legt den Sicherheitskontext fest, der beim Ausführen einer zulässigen Operation verwendet wird. Sie müssen diese Rollen kennen, bevor Sie die Basisregistry konfigurieren. Weitere Informationen zu den jeweiligen Aufgabenbereichen finden Sie unter „Rollen in der IBM MQ Console und der REST API" auf Seite 555.
- Um diese Task auszuführen, müssen Sie ein Benutzer mit ausreichenden Berechtigungen sein, um die `mqwebuser.xml`-Datei zu bearbeiten:
  - z/OS Unter z/OS müssen Sie Schreibzugriff auf die `mqwebuser.xml`-Datei haben.
  - Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

Vorgehensweise

1. Kopieren Sie die Beispiel-XML-Datei `basic_registry.xml` aus einem der folgenden Pfade:
   - ALW Unter AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
   - z/OS Unter z/OS: `PathPrefix /web/mq/samp/configuration`

     Dabei steht `PathPrefix` für den Installationspfad von IBM MQ for z/OS UNIX System Services Components.
2. Stellen Sie die Musterdatei in das entsprechende Verzeichnis:
   - ALW

     Unter AIX, Linux, and Windows: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`
   - z/OS

     Unter z/OS: *WLP_user_directory*`/servers/mqweb`

     Dabei ist *WLP_Benutzerverzeichnis* das Verzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.
3. Optional: Wenn Sie die Konfigurationseinstellungen in `mqwebuser.xml` geändert haben, kopieren Sie sie in die Musterdatei.
4. Löschen Sie die vorhandene `mqwebuser.xml` -Datei und benennen Sie die Beispieldatei in `mqwebuser.xml` um.
5. Bearbeiten Sie die neue `mqwebuser.xml` -Datei, um Benutzer und Gruppen innerhalb der **basicRegistry** -Tags hinzuzufügen.

   Beachten Sie, dass jeder Benutzer mit der Rolle " `MQWebUser` " nur die Operationen ausführen kann, die die Benutzer-ID für die Ausführung auf dem Warteschlangenmanager erteilt hat. Daher muss die in der Registry definierte Benutzer-ID über eine identische Benutzer-ID auf dem System verfügen, auf dem IBM MQ installiert ist. Diese Benutzer-IDs müssen sich in demselben Fall befinden, oder die Zuordnung zwischen den Benutzer-IDs kann fehlschlagen.

   Weitere Informationen zur Konfiguration von Basisbenutzerregistrys finden Sie im Abschnitt Basisbenutzerregistry für Liberty konfigurieren der WebSphere Liberty-Dokumentation.

6. Ordnen Sie Benutzer und Gruppen Rollen zu, indem Sie die `mqwebuser.xml` -Datei bearbeiten:

   Es sind mehrere Rollen verfügbar, die Benutzer und Gruppen berechtigen, die IBM MQ Console und REST APIzu verwenden. Jede Rolle gewährt eine andere Zugriffsebene. Weitere Informationen finden Sie unter „Rollen in der IBM MQ Console und der REST API" auf Seite 555.

   - Um Rollen zuzuweisen und Zugriff auf IBM MQ Consolezu erteilen, fügen Sie Ihre Benutzer und Gruppen zwischen den entsprechenden **security-role** -Tags innerhalb der **<enterpriseApplication id="com.ibm.mq.console">** -Tags hinzu.
   - Um Rollen zuzuweisen und Zugriff auf REST APIzu erteilen, fügen Sie Ihre Benutzer und Gruppen zwischen den entsprechenden **security-role** -Tags innerhalb der **<enterpriseApplication id="com.ibm.mq.rest">** -Tags hinzu.

   Hilfe zum Format der Benutzer-und Gruppeninformationen innerhalb der **security-role** -Tags finden Sie in den Beispielen .
7. Wenn Sie Kennwörter für Benutzer in `mqwebuser.xml`angegeben haben, sollten Sie diese Kennwörter codieren, um sie sicherer zu machen, indem Sie den von WebSphere Libertybereitgestellten Befehl **securityUtility encoding** verwenden. Weitere Informationen finden Sie im Abschnitt Liberty:securityUtility, Befehl der WebSphere Liberty-Produktdokumentation.

Beispiel

Im folgenden Beispiel wird der Gruppe `MQWebAdminGroup` Zugriff auf die IBM MQ Console mit der Rolle `MQWebAdmin` erteilt. Dem Benutzer `reader` wird der Zugriff mit der Rolle `MQWebAdminRO` erteilt, und dem Benutzer `guest` wird der Zugriff mit der Rolle `MQWebUser` erteilt:

```
<enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
            <security-role name="MQWebAdmin">
                <group name="MQWebAdminGroup" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebAdminRO">
                <user name="reader" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebUser">
                <user name="guest" realm="defaultRealm"/>
            </security-role>
        </application-bnd>
    </enterpriseApplication>
```

Im folgenden Beispiel wird den Benutzern `reader` und `guest` Zugriff auf die IBM MQ Console Konsole erteilt. Der Benutzer `user` erhält Zugriff auf die REST APIund alle Benutzer in der Gruppe `MQAdmin` erhalten Zugriff auf die IBM MQ Console und die REST API. Der `mftadmin`-Benutzer erhält Zugriff auf die REST API für MFT:

```
<enterpriseApplication id="com.ibm.mq.console">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebAdminRO">
            <user name="reader" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="guest" realm="defaultRealm"/>
        </security-role>
    </application-bnd>
</enterpriseApplication>

<enterpriseApplication id="com.ibm.mq.rest">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="user" realm="defaultRealm"/>
        </security-role>
        <security-role name="MFTWebAdmin">
            <user name="mftadmin" realm="defaultRealm"/>
        </security-role>
```

```
    </application-bnd>
</enterpriseApplication>
```

Nächste Schritte

Wählen Sie aus, wie Benutzer sich authenti

IBM MQ Console-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall gibt ein Benutzer im Anmeldebildschirm der IBM MQ Console eine Benutzer-ID und ein Kennwort ein. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Es ist keine weitere Konfiguration erforderlich, um diese Authentifizierungsoption zu verwenden, aber Sie können optional das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt Ablaufintervall für LTPA-Token konfigurieren .
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der IBM MQ Console, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden” auf Seite 561.

REST API-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die HTTP-Basisauthentifizierung verwenden In diesem Fall wird ein Benutzername und ein Kennwort codiert, aber nicht verschlüsselt, und mit jeder REST API-Anforderung gesendet, damit der Benutzer für diese Anforderung authentifiziert und berechtigt ist. Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Weitere Informationen finden Sie unter „HTTP-Basisauthentifizierung mit der REST API verwenden” auf Seite 565.
- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall stellt ein Benutzer der REST API `login`-Ressource mit der HTTP-POST-Methode eine Benutzer-ID und ein Kennwort bereit. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Weitere Informationen finden Sie unter „Tokenbasierte Authentifizierung mit der REST-API verwenden” auf Seite 566. Sie können das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der REST API, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden” auf Seite 561.

ALW Lokale OS-Registry für die IBM MQ Console und REST API konfigurieren

Sie können eine Registry des lokalen Betriebssystems in der Datei `mqwebuser.xml` konfigurieren. Die Benutzernamen und Kennwörter im lokalen Betriebssystem werden verwendet, um die Benutzer der IBM MQ Console und REST API zu authentifizieren und zu berechtigen.

Vorbereitende Schritte

- Für die Clientzertifikatsauthentifizierung mit der lokalen Betriebssystemauthentifizierungsfunktion ist die Benutzeridentität der allgemeine Name (CN) aus dem definierten Namen (DN) des Clientzertifikats. Wenn die Benutzeridentität nicht als Betriebssystembenutzer vorhanden ist, schlägt die Clientzertifikatsanmeldung fehl und wird auf die kennwortbasierte Authentifizierung zurückgetworen.
- Um diese Task ausführen zu können, müssen Sie ein privilegierter Benutzer sein.

Informationen zu diesem Vorgang

Bei einer lokalen Betriebssystemregistry wird Benutzern und Gruppen automatisch eine Rolle zugeordnet:

- Jeder Benutzer, der Teil der Gruppe 'mqm' oder der Gruppe 'QMQMADM' unter IBM i ist, erhält die Rollen 'MQWebAdmin' und 'MFTWebAdmin'.
- Allen anderen Benutzern wird die Rolle "MQWebUser" erteilt.

Weitere Informationen zu diesen Rollen finden Sie in „Rollen in der IBM MQ Console und der REST API” auf Seite 555.

Eine lokale Betriebssystemregistry kann nur unter AIX, Linux, and Windows verwendet werden. Die entsprechende Funktion wird unter z/OS bereitgestellt, indem eine SAF-Registry konfiguriert wird. Weitere Informationen finden Sie unter „SAF-Registry für die IBM MQ Console und REST API konfigurieren” auf Seite 553.

Vorgehensweise

1. Kopieren Sie die Beispiel-XML-Datei `local_os_registry.xml` aus dem folgenden Pfad:
   MQ_INSTALLATION_PATH`/web/mq/samp/configuration`
2. Stellen Sie die Beispieldatei in das folgende Verzeichnis:
   MQ_DATA_PATH`/web/installations/`*installationName*`/servers/mqweb`
3. Optional: Wenn Sie die Konfigurationseinstellungen in `mqwebuser.xml` geändert haben, kopieren Sie sie in die Musterdatei.
4. Löschen Sie die vorhandene `mqwebuser.xml` -Datei und benennen Sie die Beispieldatei in `mqwebuser.xml` um.

Nächste Schritte

Wählen Sie aus, wie Benutzer sich authenti

IBM MQ Console-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall gibt ein Benutzer im Anmeldebildschirm der IBM MQ Console eine Benutzer-ID und ein Kennwort ein. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Es ist keine weitere Konfiguration erforderlich, um diese Authentifizierungsoption zu verwenden, aber Sie können optional das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt Ablaufintervall für LTPA-Token konfigurieren .
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der IBM MQ Console, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden” auf Seite 561.

REST API-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die HTTP-Basisauthentifizierung verwenden In diesem Fall wird ein Benutzername und ein Kennwort codiert, aber nicht verschlüsselt, und mit jeder REST API-Anforderung gesendet, damit der Benutzer für diese Anforderung authentifiziert und berechtigt ist. Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Weitere Informationen finden Sie unter „HTTP-Basisauthentifizierung mit der REST API verwenden” auf Seite 565.
- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall stellt ein Benutzer der REST API `login`-Ressource mit der HTTP-POST-Methode eine Benutzer-ID und ein Kennwort bereit. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Weitere Informationen finden Sie unter „Tokenbasierte Authentifizierung mit der REST-API verwenden” auf Seite 566. Sie können

das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.

- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der REST API, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden" auf Seite 561.

LDAP-Registry für die IBM MQ Console und REST API konfigurieren

Sie können eine LDAP-Registry in der Datei `mqwebuser.xml` konfigurieren. Die Benutzernamen und Kennwörter in der LDAP-Registry werden verwendet, um Benutzer von IBM MQ Console und REST API zu authentifizieren und zu berechtigen.

Vorbereitende Schritte

- Wenn Sie eine LDAP-Registry konfigurieren, müssen Sie jedem Benutzer eine Rolle zuordnen. Jede Rolle stellt verschiedene Berechtigungsebenen für den Zugriff auf die IBM MQ Console und die REST API bereit und legt den Sicherheitskontext fest, der beim Ausführen einer zulässigen Operation verwendet wird. Sie müssen diese Rollen verstehen, bevor Sie die Registry konfigurieren. Weitere Informationen zu den jeweiligen Aufgabenbereichen finden Sie unter „Rollen in der IBM MQ Console und der REST API" auf Seite 555.

  Beachten Sie, dass jeder Benutzer mit der Rolle " `MQWebUser` " nur die Operationen ausführen kann, die die Benutzer-ID für die Ausführung auf dem Warteschlangenmanager erteilt hat. Daher muss die Benutzer-ID, die auf dem LDAP-Server definiert ist, eine identische Benutzer-ID auf dem System haben, auf dem IBM MQ installiert ist. Diese Benutzer-IDs müssen sich in demselben Fall befinden, oder die Zuordnung zwischen den Benutzer-IDs kann fehlschlagen.
- Um diese Task auszuführen, müssen Sie ein Benutzer mit ausreichenden Berechtigungen sein, um die `mqwebuser.xml`-Datei zu bearbeiten:
  - z/OS Unter z/OS müssen Sie Schreibzugriff auf die `mqwebuser.xml`-Datei haben.
  - Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

Vorgehensweise

1. Kopieren Sie die Beispiel-XML-Datei `ldap_registry.xml` aus einem der folgenden Pfade:
   - ALW Unter AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
   - z/OS Unter z/OS: `PathPrefix /web/mq/samp/configuration`

     Dabei steht `PathPrefix` für den Installationspfad von IBM MQ for z/OS UNIX System Services Components.
2. Stellen Sie die Musterdatei in das entsprechende Verzeichnis:
   - ALW

     Unter AIX, Linux, and Windows: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`
   - z/OS

     Unter z/OS: *WLP_user_directory*`/servers/mqweb`

     Dabei ist *WLP_Benutzerverzeichnis* das Verzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.

3. Optional: Wenn Sie die Konfigurationseinstellungen in `mqwebuser.xml` geändert haben, kopieren Sie sie in die Musterdatei.
4. Löschen Sie die vorhandene `mqwebuser.xml` -Datei und benennen Sie die Beispieldatei in `mqwebuser.xml` um.
5. Bearbeiten Sie die neue `mqwebuser.xml` -Datei, um die LDAP-Registry-Einstellungen in den Tags **`ldapRegistry`** und **`idsLdapFilterProperties`** zu ändern.

   Weitere Informationen zur Konfiguration von LDAP-Registrys finden Sie im Abschnitt LDAP-Benutzerregistrys in Liberty konfigurieren der WebSphere Liberty-Dokumentation.
6. Ordnen Sie Benutzer und Gruppen Rollen zu, indem Sie die `mqwebuser.xml` -Datei bearbeiten:

   Es sind mehrere Rollen verfügbar, die Benutzer und Gruppen berechtigen, die IBM MQ Console und REST APIzu verwenden. Jede Rolle gewährt eine andere Zugriffsebene. Weitere Informationen finden Sie unter „Rollen in der IBM MQ Console und der REST API” auf Seite 555.

   - Um Rollen zuzuweisen und Zugriff auf IBM MQ Consolezu erteilen, fügen Sie Ihre Benutzer und Gruppen zwischen den entsprechenden **`security-role`** -Tags innerhalb der **`<enterpriseApplication id="com.ibm.mq.console">`** -Tags hinzu.
   - Um Rollen zuzuweisen und Zugriff auf REST APIzu erteilen, fügen Sie Ihre Benutzer und Gruppen zwischen den entsprechenden **`security-role`** -Tags innerhalb der **`<enterpriseApplication id="com.ibm.mq.rest">`** -Tags hinzu.

Nächste Schritte

Wählen Sie aus, wie Benutzer sich authenti

IBM MQ Console-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall gibt ein Benutzer im Anmeldebildschirm der IBM MQ Console eine Benutzer-ID und ein Kennwort ein. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Es ist keine weitere Konfiguration erforderlich, um diese Authentifizierungsoption zu verwenden, aber Sie können optional das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt Ablaufintervall für LTPA-Token konfigurieren .
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der IBM MQ Console, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden” auf Seite 561.

REST API-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die HTTP-Basisauthentifizierung verwenden In diesem Fall wird ein Benutzername und ein Kennwort codiert, aber nicht verschlüsselt, und mit jeder REST API-Anforderung gesendet, damit der Benutzer für diese Anforderung authentifiziert und berechtigt ist. Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Weitere Informationen finden Sie unter „HTTP-Basisauthentifizierung mit der REST API verwenden” auf Seite 565.
- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall stellt ein Benutzer der REST API `login`-Ressource mit der HTTP-POST-Methode eine Benutzer-ID und ein Kennwort bereit. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Weitere Informationen finden Sie unter „Tokenbasierte Authentifizierung mit der REST-API verwenden” auf Seite 566. Sie können das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der REST API, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter

„Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden” auf Seite 561.

z/OS SAF-Registry für die IBM MQ Console und REST API konfigurieren

Über die SAF-Schnittstelle (System Authorization Facility) kann der mqweb-Server den externen Sicherheitsmanager für die Authentifizierung und die Berechtigungsprüfung aufrufen. Ein Benutzer kann sich dann bei IBM MQ Console und REST API mit einer z/OS-Benutzer-ID und einem Kennwort anmelden.

Vorbereitende Schritte

- Wenn Sie eine SAF-Registry konfigurieren, müssen Sie den Benutzern eine Rolle zuordnen. Jede Rolle stellt verschiedene Berechtigungsebenen für den Zugriff auf die IBM MQ Console und die REST API bereit und legt den Sicherheitskontext fest, der beim Ausführen einer zulässigen Operation verwendet wird. Sie müssen diese Rollen verstehen, bevor Sie die Registry konfigurieren. Weitere Informationen zu den jeweiligen Aufgabenbereichen finden Sie unter „Rollen in der IBM MQ Console und der REST API” auf Seite 555.
- Der WebSphere Liberty-Angel-Prozess muss ausgeführt werden, um die autorisierte Schnittstelle für SAF zu verwenden. Weitere Informationen finden Sie unter Autorisierte z/OS-Services in Liberty für z/OS aktivieren.
- Um diesen Task abzuschließen, müssen Sie Schreibzugriff auf die Datei `mqwebuser.xml` und die Berechtigung zum Definieren von Sicherheitsmanagerprofilen haben.

Anmerkung: V 9.2.0.25 Ab IBM MQ 9.2.0 Fix Pack 25wurde die Beispielkonfigurationsdatei `zos_saf_registry.xml` aktualisiert, um einen doppelten Eintrag safAuthorization zu entfernen.

Diese Aktualisierung behebt das Problem, dass der Fehler ICH408I auftreten kann, wenn für MQ Console unter z/OS ein Upgrade auf eine Version durchgeführt wird, die WebSphere Liberty Profile 22.0.0.12 oder höher enthält, d. h. ab IBM MQ 9.2.0 CSU 8. Mehrere safAuthorization -Anweisungen werden nicht unterstützt und können einen Fehler ICH408I verursachen, wenn Benutzer, die nicht in der Rolle MQWebAdmin oder MQWebAdminRO in der Klasse EBJROLE sind, versuchen, über MQ Consoleauf einen z/OS -Warteschlangenmanager zuzugreifen.

Der Standardwert für **racRouteLog**, der die zu protokollierenden Zugriffsversuche angibt, ist NONE. Wenn Sie einen zusätzlichen Bericht oder Datensatz für die Sicherheitsprüfung benötigen, finden Sie weitere Informationen unter SAF-Berechtigung (safAuthorization) .

Informationen zu diesem Vorgang

Die SAF-Schnittstelle ermöglicht es dem mqweb-Server, den externen Sicherheitsmanager für die Authentifizierung und Berechtigungsprüfung für die IBM MQ Console und REST API aufzurufen.

Vorgehensweise

1. Führen Sie die Schritte in Autorisierte z/OS-Services in Liberty für z/OS aktivieren aus, um Ihren mqweb-Server-Zugriff für die Verwendung von z/OS autorisierten Services zu erteilen.

   Die Beispiel-JCL zum Starten des Angel-Prozesses befindet sich in USS_ROOT/web/templates/zos/procs/bbgzangl.jcl, wobei USS_ROOT der Pfad in z/OS UNIX System Services (z/OS UNIX) ist, in dem z/OS UNIX-Komponenten installiert werden.

   Ändern Sie in bbgzangl.jcldie Anweisung SET ROOT so, dass sie auf USS_ROOT/webverweist, z. B. /usr/lpp/mqm/V9R2M0/web.

   Weitere Informationen zum Stoppen und Starten des Angel-Prozesses finden Sie im Artikel Liberty unter z/OS verwalten.
2. Führen Sie die Schritte in Liberty: Einrichten des nicht authentifizierten Benutzers der System Authorization Facility (SAF) aus, um den nicht authentifizierten Benutzer zu erstellen, der von Liberty benötigt wird.

3. Kopieren Sie die Datei `zos_saf_registry.xml` aus dem folgenden Pfad: `PathPrefix /web/mq/samp/configuration`. Dabei steht `PathPrefix` für den Installationspfad von z/OS UNIX-Komponenten.
4. Stellen Sie die Beispieldatei in das Verzeichnis *WLP_user_directory*`/servers/mqweb` , wobei *WLP_Benutzerverzeichnis* das Verzeichnis ist, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.
5. Optional: Wenn Sie zuvor Konfigurationseinstellungen in `mqwebuser.xml` geändert haben, kopieren Sie sie in die Musterdatei.
6. Löschen Sie die vorhandene `mqwebuser.xml` -Datei und benennen Sie die Beispieldatei in `mqwebuser.xml` um.
7. Passen Sie das **safCredentials** -Element in `mqwebuser.xml` an.
   a. Setzen Sie **profilePrefix** auf einen Namen, der für Ihren Liberty-Server eindeutig ist. Wenn Sie mehr als einen mqweb-Server auf einem einzigen System ausführen, müssen Sie für jeden Server einen anderen Namen auswählen, z. B. MQWEB920 und MQWEB915.
   b. Setzen Sie **unauthenticatedUser** auf den Namen des nicht authentifizierten Benutzers, der in Schritt „2” auf Seite 553 erstellt wurde.
8. Definieren Sie den mqweb-Server APPLID auf RACF.

   Der Ressourcenname APPLID ist der Wert, den Sie im Attribut **profilePrefix** in Schritt „7” auf Seite 554 angegeben haben. Im folgenden Beispiel wird der mqweb-Server APPLID in RACF definiert:

   ```
   RDEFINE APPL profilePrefix UACC(NONE)
   ```

9. Erteilen Sie allen Benutzern oder Gruppen, die auf der MQ Console oder der REST API authentifiziert werden sollen, Lesezugriff (READ) für den mqweb-Server APPLID in der APPL-Klasse.

   Sie müssen dies auch für den nicht authentifizierten Benutzer tun, der in Schritt „2” auf Seite 553 definiert ist. Im folgenden Beispiel erhält ein Benutzer Lesezugriff auf den mqweb-Server APPLID in RACF:

   ```
   PERMIT profilePrefix CLASS(APPL) ACCESS(READ) ID(userID)
   ```

10. Verwenden Sie den Befehl **SETROPTS** RACF , um die im Speicher RACLISTed APPL-Klassenprofile zu aktualisieren:

    ```
    SETROPTS RACLIST(APPL) REFRESH
    ```

11. Definieren Sie die Profile in der EJBROLE-Klasse, die erforderlich sind, um Benutzern Zugriff auf Rollen in der MQ Console und der REST API zu erteilen.

    Im folgenden Beispiel werden die Profile in RACF definiert, wobei **profilePrefix** der Wert ist, der für das Attribut **profilePrefix** in Schritt „7” auf Seite 554 angegeben wird.

```
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdminRO UACC(NONE)
```

12. Erteilen Sie Benutzern Zugriff auf Rollen in der MQ Console und der REST API.

    Erteilen Sie dazu Benutzern oder Gruppen Lesezugriff (READ) auf eines oder mehrere der Profile in der Klasse EBJROLE, die in Schritt „11” auf Seite 554 erstellt wurde. Weitere Informationen zu den Aufgabenbereichen finden Sie im Artikel „Rollen in der IBM MQ Console und der REST API” auf Seite 555.

    Im folgenden Beispiel erhält ein Benutzer Zugriff auf die MQWebAdmin-Rolle für die REST API in RACF. Dabei steht **profilePrefix** für den Wert, der für das Attribut **profilePrefix** in Schritt „7” auf Seite 554 angegeben wurde.

```
PERMIT profilePrefix.com.ibm.mq.rest.MQWebAdmin CLASS(EJBROLE) ACCESS(READ) ID(userID)
```

Ergebnisse

Sie haben die SAF-Authentifizierung für die IBM MQ Console und die REST API eingerichtet.

Nächste Schritte

Wählen Sie aus, wie Benutzer sich authenti

IBM MQ Console-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall gibt ein Benutzer im Anmeldebildschirm der IBM MQ Console eine Benutzer-ID und ein Kennwort ein. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Es ist keine weitere Konfiguration erforderlich, um diese Authentifizierungsoption zu verwenden, aber Sie können optional das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt Ablaufintervall für LTPA-Token konfigurieren .
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der IBM MQ Console, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden" auf Seite 561.

REST API-Authentifizierungsoptionen

- Lassen Sie sich Benutzer authentifizieren, indem Sie die HTTP-Basisauthentifizierung verwenden In diesem Fall wird ein Benutzername und ein Kennwort codiert, aber nicht verschlüsselt, und mit jeder REST API-Anforderung gesendet, damit der Benutzer für diese Anforderung authentifiziert und berechtigt ist. Damit diese Authentifizierung sicher ist, müssen Sie eine sichere Verbindung verwenden. Das heißt, Sie müssen HTTPS verwenden. Weitere Informationen finden Sie unter „HTTP-Basisauthentifizierung mit der REST API verwenden" auf Seite 565.
- Lassen Sie sich Benutzer authentifizieren, indem Sie die Tokenauthentifizierung In diesem Fall stellt ein Benutzer der REST API `login`-Ressource mit der HTTP-POST-Methode eine Benutzer-ID und ein Kennwort bereit. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, angemeldet zu bleiben und für einen festgelegten Zeitraum autorisiert zu sein. Weitere Informationen finden Sie unter „Tokenbasierte Authentifizierung mit der REST-API verwenden" auf Seite 566. Sie können das Ablaufintervall für das LTPA-Token konfigurieren. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.
- Lassen Sie sich Benutzer authentifizieren, indem Sie Clientzertifikate verwenden In diesem Fall verwendet der Benutzer keine Benutzer-ID und kein Kennwort für die Anmeldung an der REST API, sondern er verwendet stattdessen das Clientzertifikat. Weitere Informationen finden Sie unter „Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden" auf Seite 561.

Rollen in der IBM MQ Console und der REST API

Wenn Sie Benutzer und Gruppen für die Verwendung von IBM MQ Console oder REST APIberechtigen, müssen Sie den Benutzern und Gruppen eine der verfügbaren Rollen zuordnen: **`MQWebAdmin`**, **`MQWebAdminRO`**, **`MQWebUser`** **`MFTWebAdmin`**und **`MFTWebAdminRO`**. Jede Rolle stellt verschiedene Berechtigungsebenen für den Zugriff auf die IBM MQ Console und die REST API bereit und legt den Sicherheitskontext fest, der beim Ausführen einer zulässigen Operation verwendet wird.

Anmerkung: Mit Ausnahme der Rolle **`MQWebUser`** muss bei der Benutzer-ID die Groß-/Kleinschreibung nicht beachtet werden. Die spezifischen Voraussetzungen für diese Rolle finden Sie in „MQWebUser" auf Seite 556 .

`MQWebAdmin`
Ein Benutzer oder eine Gruppe, der bzw. die dieser Rolle zugeordnet ist, kann alle Verwaltungsoperationen ausführen und wird unter dem Sicherheitskontext der Betriebssystembenutzer-ID ausgeführt, die zum Starten des mqweb-Servers verwendet wird.

Ein Benutzer oder eine Gruppe mit dieser Rolle hat keinen Zugriff auf die folgenden REST-Services:

- Die REST API für MFT. Zur Verwendung dieser Services muss dem Benutzer oder der Gruppe auch die Rolle **MFTWebAdmin** oder **MFTWebAdminRO** zugeordnet sein.
- Die messaging REST API. Zur Verwendung der messaging REST APImuss dem Benutzer die Rolle **MQWebUser** zugewiesen sein.

MQWebAdminRO
Diese Rolle erteilt nur Zugriff auf die IBM MQ Console oder die REST API. Ein Benutzer oder eine Gruppe, der bzw. die dieser Rolle zugeordnet ist, kann die folgenden Operationen ausführen:

- Operationen in IBM MQ-Objekten wie Warteschlangen und Kanälen anzeigen und abfragen.
- Nachrichten in Warteschlangen durchsuchen.

Ein Benutzer oder eine Gruppe, der bzw. die dieser Rolle zugeordnet ist, wird unter dem Sicherheitskontext der Betriebssystembenutzer-ID ausgeführt, die zum Starten des mqweb-Servers verwendet wird.

Ein Benutzer oder eine Gruppe mit dieser Rolle hat keinen Zugriff auf die folgenden REST-Services:

- Die REST API für MFT. Zur Verwendung dieser Services muss dem Benutzer oder der Gruppe auch die Rolle **MFTWebAdmin** oder **MFTWebAdminRO** zugeordnet sein.
- Die messaging REST API. Zur Verwendung der messaging REST APImuss dem Benutzer die Rolle **MQWebUser** zugewiesen sein.

MQWebUser
Ein Benutzer oder eine Gruppe, der bzw. die dieser Rolle zugeordnet ist, kann jede Operation ausführen, die die Benutzer-ID für die Ausführung auf dem Warteschlangenmanager erteilt hat. For example:

- Operationen in IBM MQ-Objekten wie Kanälen starten und stoppen.
- Operationen in IBM MQ-Objekten wie Warteschlangen und Kanälen definieren und festlegen.
- Operationen in IBM MQ-Objekten wie Warteschlangen und Kanälen anzeigen und abfragen.
- Einreihen und Abrufen von Nachrichten mit der messaging REST API.

Ein Benutzer oder eine Gruppe, der diese Rolle zugeordnet ist, wird im Sicherheitskontext des Principals ausgeführt und kann nur die Operationen ausführen, die der Benutzer-ID für die Ausführung auf dem Warteschlangenmanager erteilt wird.

Daher muss dem Benutzer oder der Gruppe, der bzw. die in der mqweb-Benutzerregistry definiert ist, die Berechtigung innerhalb von IBM MQ erteilt werden, damit dieser Benutzer eine Operation ausführen kann. Durch die Verwendung dieser Rolle können Sie festlegen, welche Benutzer welche Art von Zugriff auf bestimmte IBM MQ-Ressourcen haben, wenn sie IBM MQ Console und REST API verwenden.

Anmerkung:

- Die maximale Länge einer Benutzer-ID, die dieser Rolle zugeordnet ist, beträgt 12 Zeichen.
- Bei der Benutzer-ID muss die gleiche Groß-/Kleinschreibung wie in der mqweb-Benutzerregistry und im IBM MQ-System verwendet werden. Wenn sich die Groß-/Kleinschreibung der Benutzer-ID unterscheidet, wird der Benutzer möglicherweise von der IBM MQ Console und der REST API authentifiziert, ist aber nicht zur Verwendung von IBM MQ-Ressourcen berechtigt.

MFTWebAdmin
Ein Benutzer oder eine Gruppe, dem/der diese Rolle zugewiesen wurde, kann alleMFTREST-Vorgänge durchführen und arbeitet im Sicherheitskontext der Benutzer-ID des Betriebssystems, die zum Starten desmqwebServers verwendet wird.

Ein Benutzer oder eine Gruppe mit dieser Rolle hat keinen Zugriff auf die IBM MQ REST API-Services. Um diese Dienste nutzen zu können, muss dem Benutzer oder der Gruppe auch die Rolle **MQWebAdmin**, **MQWebAdminRO** oder **MQWebUser** zugewiesen werden.

MFTWebAdminRO

Diese Rolle hat lediglich Lesezugriff auf die REST API für MFT . Ein Benutzer oder eine Gruppe, der bzw. die dieser Rolle zugeordnet ist, kann Leseoperationen (GET-Anforderungen) wie Listentransfer und Listenagenten ausführen.

Ein Benutzer oder eine Gruppe, der bzw. die dieser Rolle zugeordnet ist, wird unter dem Sicherheitskontext der Betriebssystembenutzer-ID ausgeführt, die zum Starten des mqweb-Servers verwendet wird.

Ein Benutzer oder eine Gruppe mit dieser Rolle hat keinen Zugriff auf die IBM MQ REST API-Services. Um diese Dienste nutzen zu können, muss dem Benutzer oder der Gruppe auch die Rolle **MQWebAdmin**, **MQWebAdminRO** oder **MQWebUser** zugewiesen werden.

Weitere Informationen zur Konfiguration von Benutzern und Gruppe für die Verwendung dieser Rollen finden Sie unter „Benutzer und Rollen konfigurieren” auf Seite 545.

Überlappende Rollen

Ein Benutzer oder eine Gruppe kann mehr als eine Rolle zugeordnet werden. Wenn ein Benutzer in dieser Situation eine Operation ausführt, wird die höchste Berechtigungsklasse, die für die Operation gilt, verwendet. Wenn beispielsweise ein Benutzer mit den Rollen **MQWebAdminRO** und **MQWebUser** eine Operation zum Abfragen der Warteschlange ausführt, wird die Rolle **MQWebAdminRO** verwendet, und der Vorgang wird im Kontext der Systembenutzer-ID ausgeführt, mit der der Webserver gestartet wurde. Wenn derselbe Benutzer eine Definitionsoperation ausführt, wird die Rolle **MQWebUser** verwendet, und der Vorgang wird im Kontext des Prinzipals ausgeführt.

ALW Ändern des Zertifikats, das Ihrem Browser von IBM MQ Console bereitgestellt wird

Sie können die IBM MQ Console so konfigurieren, dass Ihr eigenes CA-signiertes Zertifikat für Authentifizierungszwecke dargestellt wird. Dadurch wird die selbst signierte Zertifikatswarnung entfernt, die von einem Web-Browser beim Zugriff auf die IBM MQ Console-Konsole angezeigt wird.

Vorbereitende Schritte

Konfigurieren Sie Benutzer, Gruppen und Rollen, damit sie für die Verwendung der IBM MQ Console berechtigt sind. Weitere Informationen finden Sie unter „Benutzer und Rollen konfigurieren” auf Seite 545.

Informationen zu diesem Vorgang

Die Konsolensicherheit wird von einem IBM WebSphere Application Server Liberty bereitgestellt, der von Ihrer IBM MQ-Installation verwendet wird.

Wenn Sie das Zertifikat ändern möchten, das von diesem Server an Ihren Browser übergeben wird, müssen Sie Folgendes tun:

1. Fügen Sie das Zertifikat hinzu, das Sie in den Web-Server-Keystore stellen möchten.
2. Bezeichnen Sie das Zertifikat.
3. Bearbeiten Sie die `mqwebuser.xml`-Datei, um die Standardsicherheitskonfiguration zu inaktivieren.
4. Schalten Sie die eigene Sicherheitskonfiguration in der `mqwebuser.xml`-Datei ein, und geben Sie das Zertifikat an, das Sie präsentieren möchten.

Die Prozedur setzt voraus, dass Sie:

- Verwenden eines AIX, Linux, and Windows-Systems.
- Ein privilegierter Benutzer sind.

Anmerkungen:

- Im folgenden Beispiel wird ein selbst signiertes Zertifikat erstellt und verwendet, wobei Befehle verwendet werden, die auf einer Linux-Maschine ausgegeben werden, d. h. **ls** und nicht **dir** wie auf einer Windows-Maschine.
- Dies veranschaulicht Ihnen das Konzept, entfernt aber nicht die Browserwarnung.
- Um die Browserwarnung zu entfernen, müssen Sie ein CA-signiertes Zertifikat angeben.

Vorgehensweise

1. Wenn der Liberty-Server aktiv ist, stoppen Sie den Server, indem Sie den Befehl **endmqweb** in der Befehlszeile eingeben.
2. Fügen Sie Ihr Zertifikat zum Schlüsselspeicher des Liberty-Anwendungsservers hinzu, damit er das Zertifikat finden und Ihrem Web-Browser präsentieren kann.
   a) Wechseln Sie zur Schlüsselspeicherposition, indem Sie den folgenden Befehl ausgeben, und listen Sie die Ausgabe auf:

```
cd /var/mqm/web/installations/Installation1/servers/mqweb/resources/security
ls
```

   Sie sehen z. B. die folgende Ausgabe, in der der Keystore mit dem Namen `key.jks` angezeigt wird:

```
/var/mqm/web/installations/Installation1/servers/mqweb/resources/security$
ls key.jks ltpa.keys
```

   b) Erstellen Sie ein selbst signiertes Zertifikat:

   Geben Sie den folgenden Befehl aus, um ein selbst signiertes Zertifikat für Schulungszwecke zu erstellen, das dem `key.jks` mit dem Kennwort `password` hinzugefügt wird:

```
runmqckm -cert -create -db key.jks -pw password -dn
"cn=QueueManager,o=IBM,c=UK" -label myowncertificate
```

   Mit dem Flag **-dn** können Sie die Werte angeben, die in Ihrem Zertifikat angezeigt werden.

   c) Stellen Sie sicher, dass Sie das Zertifikat erfolgreich hinzugefügt haben, indem Sie den folgenden Befehl ausgeben:

```
runmqckm -cert -list -db key.jks -pw password
```

   Sie sehen z. B. die folgende Ausgabe, die zeigt, dass das Zertifikat mit dem Kennsatz versehen wurde, zusammen mit dem Zertifikat, das mit der Bezeichnung `default` gekennzeichnet ist, die der Server derzeit verwendet:

```
/var/mqm/web/installations/Installation1/servers/mqweb/resources/security
$ runmqckm -cert -list -db key.jks -pw password
Certificates in database /var/mqm/web/installations/Installation1/servers/mqweb/resour-
ces/security/key.jks
   default
   myown certificate
```

3. Bearbeiten Sie die `mqwebuser.xml`-Datei, damit der Server das neue Zertifikat bereitstellt.
   a) Wechseln Sie in die Position der `mqwebuser.xml`-Datei und öffnen Sie sie dann zur Bearbeitung in einem Texteditor Ihrer Wahl, in diesem Fall *Nano*

```
cd /var/mqm/web/installations/Installation1/servers/mqweb
nano mqwebuser.xml
```

   b) Schalten Sie die Standardsicherheitskonfiguration aus.

   Kommentieren Sie die folgende Zeile aus, indem Sie `<!--` zum Anfang der Codezeile und `-->` am Ende der Codezeile hinzufügen:

```
<!--
```

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
-->
```

c) Aktivieren Sie Ihre eigene Konfiguration und geben Sie sie an.

Führen Sie dazu die folgende Prozedur aus:

i) Entfernen Sie die Kommentarzeichen für die folgenden Codezeilen, indem Sie die `<!--` vom Anfang des Codeblocks und `-->` vom Ende des Codeblocks entfernen.

```
<!--
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
<ssl id="thisSSLConfig" clientAuthenticationSupported="true" keyStoreRef="defaultKeyStore" serverKeyAli
as="default" trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"/>
<sslDefault sslRef="thisSSLConfig"/>
-->
```

ii) **Ändern Sie nicht die erste Zeile** des Codeblocks, da diese Zeile den Schlüsselspeicher angibt, den die Konsole verwendet, um ihre persönlichen Zertifikate zu speichern.

iii) **Kommentieren Sie die zweite Zeile des Codeblocks aus**, da diese Zeile einen Truststore angibt, in dem die Konsole nach Clientzertifikaten suchen würde. Da Sie die Tokenauthentifizierung verwenden, haben Sie keinen Truststore erstellt, und würden Sie die Codezeile nicht auskommentieren, würde dies zu einem Fehler führen, wenn die Konsole gestartet wird.

iv) **Ändern Sie serverKeyAlias="default" in serverKeyAlias="myowncertificate"** in der dritten Zeile des Codeblocks und lassen Sie alles andere gleich.

v) **Ändern Sie nicht die letzte Zeile** des Codeblocks, da diese Zeile dem Server mitteilt, die Konfiguration zu verwenden, die Sie gerade angegeben haben.

Der Codeblock sieht nun wie folgt aus:

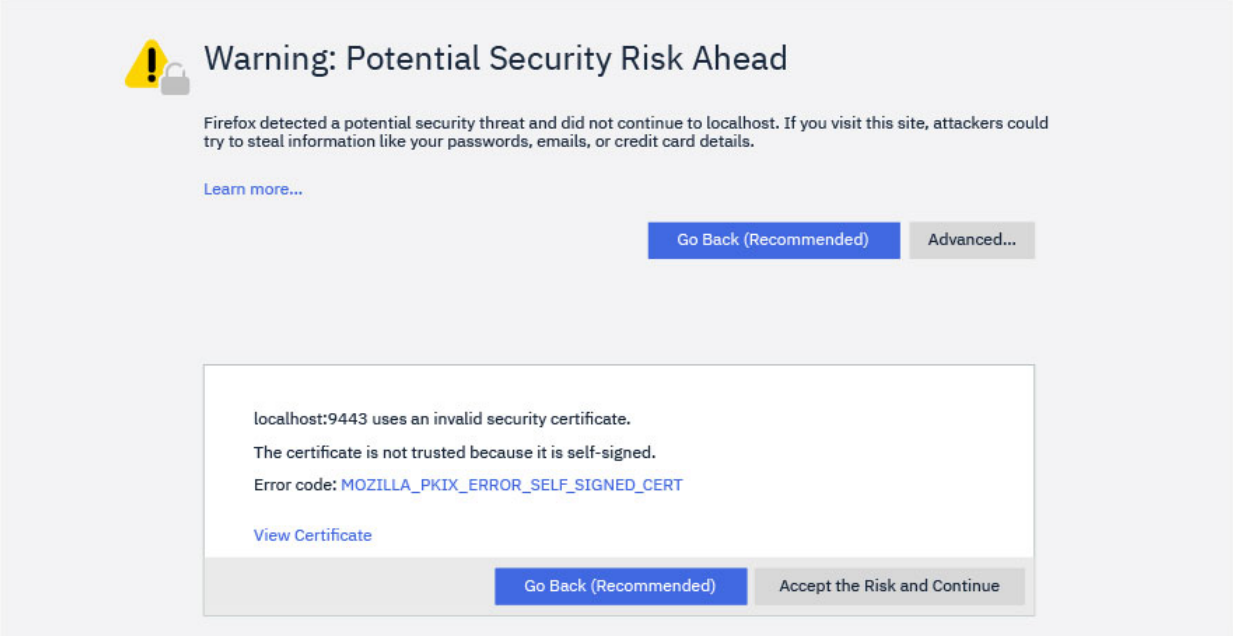
```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
<!-- Commenting out the defaultTrustStore as otherwise we get errors (viewable in the messages.log file
in the logs folder) j
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
-->
<ssl id="thisSSLConfig" clientAuthenticationSupported="true" keyStoreRef="defaultKeyStore" serverKeyAli
as="myowncertificate" trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"/>
<sslDefault sslRef="thisSSLConfig"/>
```

4. Starten Sie den Web-Server mit dem Befehl **strmqweb** erneut.

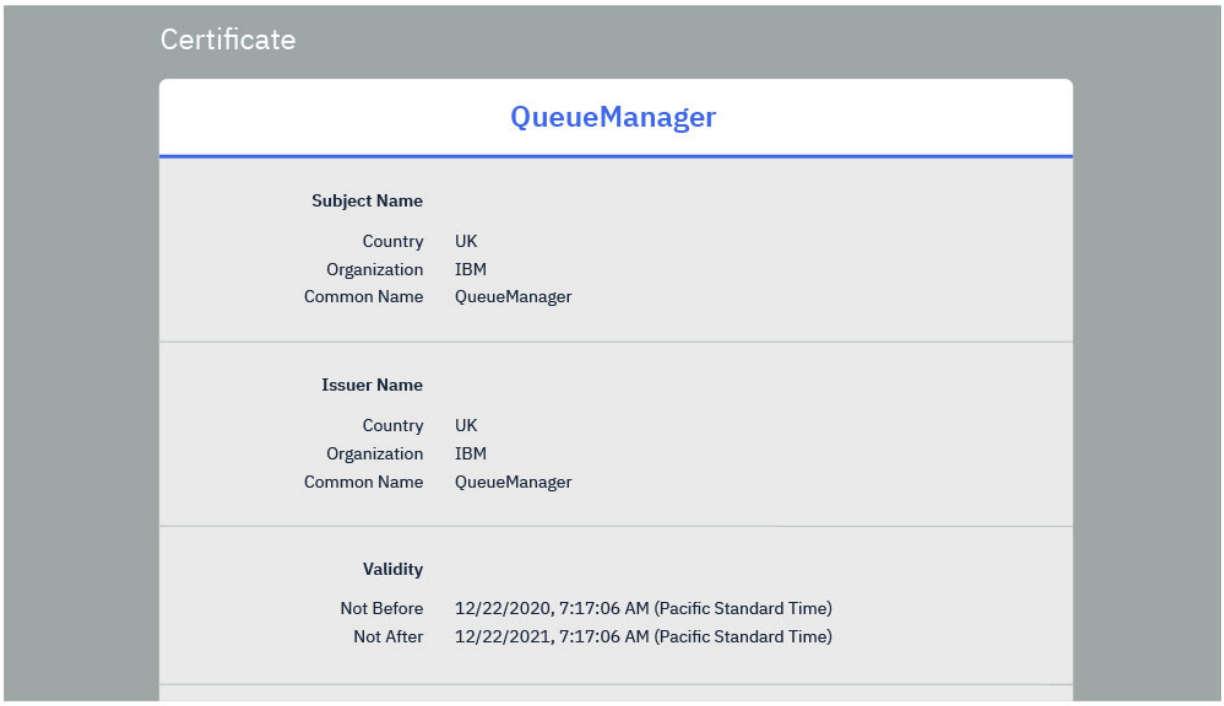
Ergebnisse

Wenn der Web-Server gestartet wird, navigieren Sie zu Ihrer IBM MQ Console und nehmen Sie eine Aktualisieren vor. Wenn Sie ein selbst signiertes Zertifikat verwenden, das Sie unter Verwendung der im vorherigen Text beschriebenen Prozedur in den Schritten „2” auf Seite 558 und „3” auf Seite 558 erstellt haben, sehen Sie eine Sicherheitswarnung.

Beachten Sie, dass das Format dieser Warnung vom verwendeten Browser abhängt.



Wenn Sie auf **Zertifikat anzeigen** klicken, sehen Sie, dass es die Details enthält, die Sie im Flag **-dn** angegeben haben, als Sie das Zertifikat in Schritt „2.b" auf Seite 558 erstellt haben.



Wenn Sie jedoch ein CA-signiertes Zertifikat verwenden, dem Ihr Browser vertraut und das Sie durch folgenden Befehl hinzugefügt haben:

```
runmqckm -cert -add -db key.jks -pw password -label myCACertificate
```

Dabei steht `myCACertificate` für den Dateipfad zu der Datei mit Ihrem CA-Zertifikat, die Sie direkt zur Anmeldeseite führen.

Achtung: Wenn Sie ein CA-signiertes Zertifikat verwenden und dieses CA-Zertifikat Teil einer Zertifikatskette ist, müssen Sie alle Zertifikate zur Kette hinzufügen, beginnend mit dem Root-CA-

Zertifikat. Weitere Informationen finden Sie unter „CA-Zertifikat oder öffentlichen Teil eines selbst signierten Zertifikats unter AIX, Linux, and Windows einem Schlüsselrepository hinzufügen" auf Seite 327.

ALW Clientzertifikatsauthentifizierung mit der REST API und der IBM MQ Console verwenden

Sie können Clientzertifikate zu Principals zuordnen, um IBM MQ Console- und REST API-Benutzer zu authentifizieren.

Vorbereitende Schritte

- Konfigurieren Sie Benutzer, Gruppe und Rollen, damit sie zur Verwendung der IBM MQ Console und REST API berechtigt sind. Weitere Informationen finden Sie unter „Benutzer und Rollen konfigurieren" auf Seite 545.
- Bei der Verwendung der REST API können Sie die Berechtigungsnachweise des aktuellen Benutzers anfordern, indem Sie die HTTP-GET-Methode in der `login`-Ressource verwenden und dem Clientzertifikat die Authentifizierung der Anforderung bereitstellen. Diese Anforderung gibt Informationen über den Benutzernamen und die Rollen zurück, denen der Benutzer zugeordnet ist. Weitere Informationen finden Sie unter GET `/login`.
- Wenn Sie Clientzertifikaten Principals zuordnen, um Benutzer zu authentifizieren, wird der definierte Name des Clientzertifikats für die Übereinstimmung mit Benutzern in der konfigurierten Benutzerregistry verwendet:
  - Für eine Basisregistry wird der allgemeine Name (Common Name, CN) mit dem Benutzer verglichen. Beispiel: `CN=Fred, O=IBM, C=GB` wird mit dem Benutzernamen `Fred` abgeglichen.
  - Bei einer LDAP-Registry wird der vollständige definierte Name standardmäßig mit LDAP abgeglichen. Sie können Filter und Zuordnungen einrichten, um den Abgleich anzupassen. Weitere Informationen finden Sie unter Liberty:LDAP-Zertifikatszuordnungsmodus in der Dokumentation zu WebSphere Liberty.

Informationen zu diesem Vorgang

Wenn sich ein Benutzer durch die Verwendung eines Clientzertifikats authentifiziert, wird das Zertifikat an Stelle eines Benutzernamens und eines Kennworts verwendet. Für den REST API wird das Clientzertifikat mit jeder REST-Anforderung zur Authentifizierung des Benutzers bereitgestellt. Wenn sich ein Benutzer auf der IBM MQ Console mit einem Zertifikat anmeldet, kann der Benutzer anschließend nicht abgemeldet werden.

Die Prozedur setzt die folgenden Informationen voraus:

- Die Datei `mqwebuser.xml` basiert auf einem der folgenden Beispiele:
  - `basic_registry.xml`
  - `local_os_registry.xml`
  - `ldap_registry.xml`
- Sie verwenden ein AIX, Linux, and Windows-System.
- Sie sind ein privilegierter Benutzer.

Um die Clientzertifikatsauthentifizierung mit einem RACF-Schlüsselring unter z/OS zu konfigurieren, befolgen Sie das Verfahren in „TLS für REST API und IBM MQ Console auf z/OS konfigurieren" auf Seite 574.

Anmerkung: In der folgenden Prozedur werden die Schritte beschrieben, die für die Verwendung der Clientzertifikate mit der IBM MQ Console und der REST API erforderlich sind. Die Schritte zum Erstellen und Verwenden von selbst signierten Zertifikaten für den Entwickler sind detailliert beschrieben. Verwenden Sie jedoch für die Produktion Zertifikate, die von einer Zertifizierungsstelle bezogen werden.

Vorgehensweise

1. Starten Sie den mqweb-Server, indem Sie den Befehl **strmqweb** in der Befehlszeile eingeben.
2. Erstellen Sie ein Clientzertifikat:
   a) Erstellen Sie einen PKCS#12-Keystore:
      i) Öffnen Sie das Tool IBM Key Management, indem Sie den Befehl **strmqikm** in der Befehlszeile eingeben.
      ii) Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) im Tool IBM Key Management auf **New** (Neu).
      iii) Wählen Sie in der Liste **Schlüsseldatenbanktyp** die Option **PKCS12** aus.
      iv) Wählen Sie eine Position zum Speichern des Keystores aus, und geben Sie einen geeigneten Namen in das Feld **Dateiname** ein. Zum Beispiel,`user.p12`
      v) Legen Sie ein Kennwort fest, wenn Sie dazu aufgefordert
   b) Erstellen Sie das Zertifikat, indem Sie entweder ein selbst signiertes Zertifikat erstellen oder ein Zertifikat von einer Zertifizierungsstelle erhalten:
   - Erstellen Sie ein selbst signiertes Zertifikat:
      i) Klicken Sie auf **New Self-Signed** .
      ii) Geben Sie `user` in das Feld **Key Label** ein.
      iii) Wenn Sie eine Basisbenutzerregistry verwenden, geben Sie den Namen eines Benutzers aus Ihrer Benutzerregistry in das Feld **Common Name** ein. Beispiel: `mqadmin`. Stellen Sie für eine LDAP-Benutzerregistry sicher, dass der definierte Name für das Zertifikat mit dem definierten Namen in der LDAP-Registry übereinstimmt.
      iv) Klicken Sie auf **OK**.
   - Fordern Sie ein Zertifikat von einer Zertifizierungsstelle an. Das CA-Zertifikat muss den entsprechenden Benutzernamen innerhalb des allgemeinen Namens (CN) des Felds für den registrierten Namen (DN) enthalten:
      i) Fordern Sie ein neues Zertifikat an. Klicken Sie im Menü **Erstellen** auf **Neue Zertifikatsanforderung** .
      ii) Geben Sie in das Feld **Schlüsselkennsatz** die Zertifikatsbezeichnung ein.
      iii) Wenn Sie eine Basisbenutzerregistry verwenden, geben Sie in das Feld **Allgemeiner Name** den Benutzernamen des Benutzers ein, für den das Zertifikat gilt.

      Wenn Sie eine lokale Betriebssystemregistry verwenden, muss das Feld **Common Name** mit der lokalen Betriebssystembenutzer-ID übereinstimmen.

      Stellen Sie für eine LDAP-Benutzerregistry sicher, dass der definierte Name für das Zertifikat mit dem definierten Namen in der LDAP-Registry übereinstimmt.
      iv) Geben Sie die Werte für die übrigen Felder ein, oder wählen Sie sie aus.
      v) Wählen Sie aus, wo die Zertifikatsanforderung gespeichert werden soll, und wählen Sie den Dateinamen für die Zertifikatsanforderung aus und klicken Sie dann auf **OK** .
      vi) Senden Sie die Zertifikatsanforderungsdatei an eine Zertifizierungsstelle (Certificate Authority, CA).
      vii) Wenn Sie über das Zertifikat von der Zertifizierungsstelle verfügen, öffnen Sie das Tool IBM Key Management, indem Sie den Befehl **strmqikm** in der Befehlszeile eingeben.
      viii) Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) im Tool IBM Key Management auf **Open** .
      ix) Wählen Sie den PKCS#12-Keystore aus, der das Clientzertifikat enthält. Beispiel:`user.p12`
      x) Klicken Sie auf **Empfangen** , wählen Sie das entsprechende Zertifikat aus und klicken Sie auf **OK** .
3. Extrahieren Sie den öffentlichen Teil des Clientzertifikats:

a) Öffnen Sie das Tool IBM Key Management, indem Sie den Befehl **strmqikm** in der Befehlszeile eingeben.

b) Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) im Tool IBM Key Management auf **Open** .

c) Wählen Sie den PKCS#12-Keystore aus, der das Clientzertifikat enthält. Beispiel:`user.p12`

d) Wählen Sie das Clientzertifikat aus der Zertifikatsliste im IBM Key Management-Tool aus.

e) Klicken Sie auf **Extract Certificate** (Zertifikat extrahieren

f) Wählen Sie eine Position zum Speichern des Zertifikats aus, und geben Sie einen geeigneten Dateinamen in das Feld **Name der Zertifikatsdatei** ein. Beispiel: `user.arm`.

4. Importieren Sie den öffentlichen Teil des Clientzertifikats in den Trust-Keystore des mqweb-Servers als Unterzeichnerzertifikat, damit der Server das Clientzertifikat überprüfen kann:

a) Erstellen Sie einen `trust.jks` -Keystore für die Verwendung durch den mqweb-Server, falls noch keiner vorhanden ist:

i) Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) im Tool IBM Key Management auf **New** (Neu).

ii) Wählen Sie **JKS** in der Liste **Schlüsseldatenbanktyp** aus.

iii) Klicken Sie auf **Durchsuchen** und navigieren Sie zu `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb/resources/security`.

Dieses Verzeichnis sollte bereits eine `key.jks` -Datei enthalten. Wenn bereits eine `trust.jks` -Datei vorhanden ist, öffnen Sie die vorhandene Datei, und überschreiben Sie sie nicht.

iv) Geben Sie `trust.jks` in das Feld **Dateiname** ein.

v) Legen Sie ein Kennwort fest, wenn Sie dazu aufgefordert

b) Wählen Sie im Dropdown-Menü **Signer Certificates** (Unterzeichnerzertifikate) aus.

c) Klicken Sie auf **Hinzufügen**.

d) Wählen Sie die entsprechende Armdatei aus, und klicken Sie auf **OK** . Wählen Sie z. B. `user.arm` aus.

e) Geben Sie eine Bezeichnung für das Zertifikat ein.

5. Ändern Sie das Kennwort für den mqweb-Server-Keystore:

a) Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Open** (Öffnen).

b) Wählen Sie **JKS** in der Liste **Schlüsseldatenbanktyp** aus.

c) Klicken Sie auf **Durchsuchen** und navigieren Sie zu *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb/resources/security`

d) Wählen Sie den `key.jks` -Keystore aus und klicken Sie auf **Öffnen** .

e) Geben Sie das Kennwort ein, wenn Sie dazu aufgefordert Das Standardkennwort ist `password`.

f) Klicken Sie im Menü **Key Database File** (Schlüsseldatenbankdatei) auf **Change Password** (Kennwort ändern)

g) Geben Sie ein neues Kennwort für den Keystore ein.

6. Aktivieren Sie die Clientzertifikatsauthentifizierung in der `mqwebuser.xml`-Datei:

Die Datei `mqwebuser.xml` kann im folgenden Pfad gefunden werden: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`

a) Entfernen Sie die Kommentarzeichen für den Abschnitt in der `mqwebuser.xml`-Datei, der die Clientzertifikatsauthentifizierung aktiviert. Der Abschnitt enthält den folgenden Text:

```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
    <keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
    <ssl id="thisSSLConfig" clientAuthenticationSupported="true" keyStoreRef="defaultKey
Store"
            trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2" serverKeyAlias="de
```

```
fault"/>
    <sslDefault sslRef="thisSSLConfig"/>
```

b) Überprüfen Sie, ob der Wert **serverKeyAlias** mit dem Namen des Serverzertifikats übereinstimmt. Wenn Sie das Standardserverzertifikat verwenden, ist der Wert korrekt.

c) Ändern Sie den Wert für das **Kennwort** für den `defaultKeyStore` in eine verschlüsselte Version des Kennworts für den `key.jks` -Keystore:

i) Geben Sie in das Verzeichnis *MQ_INSTALLATION_PATH*`/web/bin` den folgenden Befehl in der Befehlszeile ein:

```
securityUtility encode password
```

ii) Stellen Sie die Ausgabe dieses Befehls in das Feld **Kennwort** für den `defaultKeyStore`.

d) Ändern Sie den Wert für **Kennwort** für den `defaultTrustStore` so, dass er mit dem Kennwort für den `trust.jks` -Keystore übereinstimmt:

i) Geben Sie in das Verzeichnis *MQ_INSTALLATION_PATH*`/web/bin` den folgenden Befehl in der Befehlszeile ein:

```
securityUtility encode password
```

ii) Stellen Sie die Ausgabe dieses Befehls in das Feld **Kennwort** für den `defaultTrustStore`.

e) Entfernen Sie die folgende Zeile aus der `mqwebuser.xml`-Datei oder setzen Sie diese auf Kommentar:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

7. Stoppen Sie den mqweb-Server, indem Sie den Befehl **endmqweb** in der Befehlszeile eingeben.
8. Starten Sie den mqweb-Server, indem Sie den Befehl **strmqweb** in der Befehlszeile eingeben.
9. Verwenden Sie das Clientzertifikat zur Authentifizierung:
   - Wenn Sie das Clientzertifikat mit der IBM MQ Console verwenden möchten, installieren Sie das Clientzertifikat in dem Web-Browser, mit dem auf die IBM MQ Console zugegriffen wird. Installieren Sie zum Beispiel das Clientzertifikat `user.p12` als persönliches Zertifikat.
   - Wenn Sie das Clientzertifikat mit der REST API verwenden möchten, stellen Sie das Clientzertifikat mit jeder REST-Anforderung bereit. Wenn Sie HTTP POST-, PATCH- oder DELETE-Methoden verwenden, müssen Sie eine zusätzliche Authentifizierung mit dem Clientzertifikat bereitstellen, um zu verhindern, dass Cross-Site-Request-Forgery-Attacken durchgeführt werden. Dies bedeutet, dass die zusätzliche Authentifizierung verwendet wird, um zu bestätigen, dass die Berechtigungsnachweise, die für die Authentifizierung der Anforderung verwendet werden, vom Eigner der Berechtigungsnachweise verwendet werden.

     Diese zusätzliche Authentifizierung wird durch den HTTP-Header `ibm-mq-rest-csrf-token` bereitgestellt. Setzen Sie den Wert des Headers `ibm-mq-csrf-token` auf alles, was auch leer ist, und übergeben Sie die Anforderung.

Beispiel

Wichtig: In dem Beispiel unterstützen nicht alle cURL-Implementierungen selbst signierte Zertifikate, daher müssen Sie eine cURL-Implementierung verwenden, die dies tut.

Das folgende cURL-Beispiel zeigt, wie eine neue Warteschlange `Q1` auf WS-Manager `QM1` mit Clientzertifikatsauthentifizierung erstellt wird. Die genaue Konfiguration dieses cURL-Befehls hängt von den Bibliotheken ab, für die cURL erstellt wurde. Das Beispiel basiert auf einem Windows-System, wobei cURL für OpenSSL erstellt wird.

- Verwenden Sie die HTTP-POST-Methode mit der Warteschlangenressource und authentifizieren Sie sich mit dem Clientzertifikat und einschließlich des HTTP-Headers `ibm-mq-rest-csrf-token` mit einem beliebigen Wert. Dieser Wert kann alles sein, einschließlich Leerzeichen. Das Flag `--cert-type`

gibt an, dass es sich bei dem Zertifikat um ein PKCS#12-Zertifikat handelt. Das Flag `--cert` gibt die Position des Zertifikats, gefolgt von einem Doppelpunkt,: und dann das Kennwort für das Zertifikat an:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -
-cert-type P12 --cert c:\user.p12:password
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

HTTP-Basisauthentifizierung mit der REST API verwenden

Benutzer der REST API können sich authentifizieren, indem Sie ihre Benutzer-ID und das zugehörige Kennwort in einem HTTP-Header bereitstellen. Um diese Methode der Authentifizierung mit HTTP-Methoden wie POST, PATCH und DELETE verwenden zu können, müssen auch der HTTP-Header `ibm-mq-rest-csrf-token` sowie eine Benutzer-ID und ein Kennwort angegeben werden.

Vorbereitende Schritte

- Konfigurieren Sie Benutzer, Gruppen und Rollen, damit sie für die Verwendung der REST API berechtigt sind. Weitere Informationen finden Sie unter „Benutzer und Rollen konfigurieren” auf Seite 545.
- Stellen Sie sicher, dass die HTTP-Basisauthentifizierung aktiviert ist. Überprüfen Sie, ob die folgende XML vorhanden ist und in der Datei `mqwebuser.xml` nicht auf Kommentar gesetzt ist. Diese XML muss in den `<featureManager>`-Tags enthalten sein:

  ```
  <feature>basicAuthenticationMQ-1.0</feature>
  ```

  z/OS Unter z/OS müssen Sie ein Benutzer sein, der über Schreibzugriff auf `mqwebuser.xml` verfügt, um diese Datei zu bearbeiten.

  Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein, um die `mqwebuser.xml`-Datei zu bearbeiten.
- Stellen Sie sicher, dass Sie eine sichere Verbindung verwenden, wenn Sie REST-Anforderungen senden. Da die Kombination aus Benutzername und Kennwort codiert, aber nicht verschlüsselt ist, müssen Sie bei der Verwendung der HTTP-Basisauthentifizierung mit der REST API eine sichere Verbindung (HTTPS) verwenden.
- Sie können die Berechtigungsnachweise des aktuellen Benutzers abfragen, indem Sie die HTTP GET-Methode in der `login` -Ressource verwenden und die Basisauthentifizierungsinformationen zur Authentifizierung der Anforderung bereitstellen. Diese Anforderung gibt Informationen über den Benutzernamen und die Rollen zurück, denen der Benutzer zugeordnet ist. Weitere Informationen finden Sie unter GET `/login`.

Vorgehensweise

1. Konkatenieren Sie den Benutzernamen mit einem Doppelpunkt und das Kennwort. Beachten Sie, dass bei dem Benutzernamen die Groß-/Kleinschreibung beachtet werden muss.

   Beispiel: Der Benutzername 'admin' und das Kennwort 'admin' werden zur folgenden Zeichenfolge:

   ```
   admin:admin
   ```
2. Codieren Sie diesen Benutzernamen und die zugehörige Kennwortzeichenfolge in base64-Codierung.
3. Geben Sie diesen codierten Benutzernamen und das Kennwort in einem HTTP-Header `Authorization: Basic` an.

   Wenn Sie beispielsweise einen verschlüsselten Benutzernamen mit Administratorberechtigung und ein Kennwort für admin verwenden, wird der folgende Header erstellt:

   ```
   Authorization: Basic YWRtaW46YWRtaW4=
   ```
4. Wenn Sie HTTP-POST-, PATCH-oder DELETE-Methoden verwenden, müssen Sie zusätzliche Authentifizierung sowie einen Benutzernamen und ein Kennwort bereitstellen.

Diese zusätzliche Authentifizierung wird durch den HTTP-Header `ibm-mq-rest-csrf-token` bereitgestellt. Der HTTP-Header `ibm-mq-rest-csrf-token` muss in der Anforderung vorhanden sein, aber sein Wert kann alles sein, einschließlich Leerzeichen.

5. Übergeben Sie Ihre REST-Anforderung mit den entsprechenden Headern an IBM MQ.

Beispiel

Im folgenden Beispiel wird gezeigt, wie die neue Warteschlange `Q1` im Warteschlangenmanager QM1 mit der Basisauthentifizierung auf Windows-Systemen erstellt wird. Im Beispiel wird cURL verwendet:

- Verwenden Sie die HTTP-POST-Methode mit der Warteschlangenressource, die sich mit der Basisauthentifizierung authentifiziert und den HTTP-Header `ibm-mq-rest-csrf-token` mit einem beliebigen Wert enthält. Dieser Wert kann wie folgt angegeben werden:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST
-u mqadmin:mqadmin
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

Tokenbasierte Authentifizierung mit der REST-API verwenden

Benutzer der REST API können sich authentifizieren, indem Sie der REST API-Ressource `login` mit der HTTP POST-Methode eine Benutzer-ID und ein Kennwort bereitstellen. Es wird ein LTPA-Token generiert, das es dem Benutzer ermöglicht, zukünftige Anforderungen zu authentifizieren. Dieses LTPA-Token hat das Präfix `LtpaToken2`. Der Benutzer kann sich mit der HTTP-Methode DELETE abmelden und kann das Protokoll in Informationen des aktuellen Benutzers mit der HTTP GET-Methode abfragen.

Vorbereitende Schritte

- Konfigurieren Sie Benutzer, Gruppen und Rollen, damit sie für die Verwendung der REST API berechtigt sind. Weitere Informationen finden Sie unter „Benutzer und Rollen konfigurieren” auf Seite 545.
- Standardmäßig beginnt der Name des Cookies, das das LTPA-Token enthält, mit `LtpaToken2` und enthält ein Suffix, das sich ändern kann, wenn der mqweb-Server erneut gestartet wird. Dieser randomisierte Cookiename ermöglicht es, dass mehr als ein mqweb-Server auf demselben System ausgeführt wird. Wenn der Cookiename jedoch ein konsistenter Wert bleiben soll, können Sie den Namen des Cookies mit dem Befehl **setmqweb** angeben. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.
- Standardmäßig läuft das LTPA-Token-Cookie nach 120 Minuten ab. Sie können die Ablaufzeit des LTPA-Token-Cookies mit dem Befehl **setmqweb** konfigurieren. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.
- Stellen Sie sicher, dass Sie eine sichere Verbindung verwenden, wenn Sie REST-Anforderungen senden. Wenn Sie die HTTP-POST-Methode in der `login` -Ressource verwenden, werden die Kombination aus Benutzername und Kennwort, die mit der Anforderung gesendet wird, nicht verschlüsselt. Daher müssen Sie bei der Verwendung der tokenbasierten Authentifizierung mit der REST API eine sichere Verbindung (HTTPS) verwenden. Standardmäßig können Sie HTTP nicht mit der LTPA-Tokenauthentifizierung verwenden. Sie können das LTPA-Token aktivieren, das von insicheren HTTP-Verbindungen verwendet werden soll, indem Sie **secureLTPA** auf `False` setzen. Weitere Informationen finden Sie im Abschnitt LTPA-Token konfigurieren.
- Sie können die Berechtigungsnachweise des aktuellen Benutzers abfragen, indem Sie die HTTP GET-Methode in der `login` -Ressource verwenden und das LTPA-Token für die Authentifizierung der Anforderung bereitstellen. Diese Anforderung gibt Informationen über den Benutzernamen und die Rollen zurück, denen der Benutzer zugeordnet ist. Weitere Informationen finden Sie unter GET `/login` .

Vorgehensweise

1. Melden Sie sich an einem Benutzer an:

   a) Verwenden Sie die HTTP-POST-Methode in der `login` -Ressource:

```
https://host:port/ibmmq/rest/v1/login
```

Geben Sie den Benutzernamen und das Kennwort im Hauptteil der JSON-Anforderung in das folgende Format ein:

```
{
    "username" : name,
    "password" : password
}
```

b) Speichern Sie das LTPA-Token, das von der Anforderung im lokalen Cookiespeicher zurückgegeben wird. Dieses LTPA-Token hat standardmäßig das Präfix `LtpaToken2`.

2. Authentifizieren Sie REST-Anforderungen mit dem gespeicherten LTPA-Token als ein Cookie mit jeder Anforderung.

   Für Anforderungen, die die Methoden HTTP PUT, PATCH oder DELETE verwenden, schließen Sie einen `ibm-mq-rest-csrf-token` -Header ein. Der Wert dieses Headers kann alles sein, einschließlich leer.

3. Melden Sie einen Benutzer ab:

   a) Verwenden Sie die HTTP-Methode DELETE für die `login` -Ressource:

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

   Sie müssen das LTPA-Token als Cookie bereitstellen, um die Anforderung zu authentifizieren, und einen `ibm-mq-rest-csrf-token` -Header enthalten. Der Wert dieses Headers kann alles sein, einschließlich Leerzeichen.

   b) Verarbeiten Sie die Anweisung, um das LTPA-Token aus dem lokalen Cookie-Speicher zu löschen.

   Anmerkung: Wenn die Anweisung nicht verarbeitet wird und das LTPA-Token im lokalen Cookiespeicher verbleibt, kann das LTPA-Token verwendet werden, um zukünftige REST-Anforderungen zu authentifizieren. Das heißt, wenn der Benutzer versucht, nach Beendigung der Sitzung mit dem LTPA-Token zu authentifizieren, wird eine neue Sitzung erstellt, die das vorhandene Token verwendet.

Beispiel

Das folgende cURL-Beispiel zeigt, wie die neue Warteschlange `Q1` im Warteschlangenmanager `QM1` mit der tokenbasierten Authentifizierung auf Windows-Systemen erstellt wird:

- Melden Sie sich an, und fügen Sie das LTPA-Token mit dem Präfix `LtpaToken2` zum lokalen Cookiespeicher hinzu. Die Informationen zu Benutzername und Kennwort sind im JSON-Hauptteil enthalten. Das Flag `-c` gibt die Position der Datei an, in der das Token gespeichert werden soll:

  ```
  curl -k https://localhost:9443/ibmmq/rest/v1/login -X POST
  -H "Content-Type: application/json" --data "{\"username\":\"mqadmin\",\"password\":\"mqad
  min\"}"
  -c c:\cookiejar.txt
  ```

- Erstellen Sie eine Warteschlange. Verwenden Sie die HTTP-POST-Methode mit der Warteschlangenressource, die mit dem LTPA-Token authentifiziert wird. Das LTPA-Token mit dem Präfix `LtpaToken2` wird mit dem Flag `-b` aus der Datei `cookiejar.txt` abgerufen. Der CSRF-Schutz wird durch das Vorhandensein des HTTP-Headers `ibm-mq-rest-csrf-token` bereitgestellt:

  ```
  curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -b
  c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json"
  --data "{\"name\":\"Q1\"}"
  ```

- Melden Sie sich ab und löschen Sie das LTPA-Token aus dem lokalen Cookie-Store. Das LTPA-Token wird unter Verwendung der Markierung `-b` aus der Datei `cookiejar.txt` abgerufen. Der CSRF-Schutz wird durch das Vorhandensein des HTTP-Headers `ibm-mq-rest-csrf-token` bereitgestellt. Die Posi-

tion der Datei `cookiejar.txt` wird durch die Markierung `-c` angegeben, so dass das LTPA-Token aus der Datei gelöscht wird:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X DELETE
-H "ibm-mq-rest-csrf-token: value" -b c:\cookiejar.txt
-c c:\cookiejar.txt
```

Zugehörige Verweise

POST /login

GET/login

/login löschen

V 9.2.0 Integration der IBM MQ Console in einen I-Frame

Mit dem HTML-Element <iframe> kann eine Webseite mithilfe eines Informationsrahmens (I-Frame) in eine andere eingebettet werden. Aus Sicherheitsgründen kann die IBM MQ Console standardmäßig nicht in einen I-Frame eingebettet werden. Sie können jedoch einen I-Frame aktivieren, indem Sie die Konfigurationseigenschaft **mqConsoleFrameAncestors** auf dem mqweb-Server verwenden.

Informationen zu diesem Vorgang

Der mqweb-Server verwaltet eine Zulassungsliste mit den Ursprüngen von Webseiten, die die IBM MQ Console unter Verwendung eines I-Frame einbetten können. Ein Ursprung ist eine Kombination aus einem URL-Schema, einer Domäne und einem Port, zum Beispiel `https://example.com:1234`.

Sie können mithilfe der Konfigurationseigenschaft **mqConsoleFrameAncestors** auf dem mqweb-Server die Einträge in der Liste angeben.

Standardmäßig ist **mqConsoleFrameAncestors** leer, d. h., die IBM MQ Console kann nicht in einen I-Frame eingebettet werden.

Vorgehensweise

Geben Sie eine Liste mit Ursprüngen von Webseiten an, die die IBM MQ Console in einen I-Frame einbetten kann. Geben Sie hierzu den folgenden Befehl ein:

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

Dabei ist *allowedOrigins* eine durch Kommas getrennte Liste mit Ursprüngen. Jeder Ursprung sollte folgende Bestandteile umfassen:

- Ein Hostname oder eine IP-Adresse
- Ein optionales URL-Schema
- Eine optionale Portnummer

Beachten Sie, dass der Hostname mit dem Platzhalterzeichen (*) beginnen kann; ebenso kann auch die Portnummer kann auch das Platzhalterzeichen (*) verwenden.

Beispiel-Ursprünge:

```
https://example.com:1234
```

Ermöglicht jeder Webseite, die von `https://example.com:1234` bereitgestellt wird, die IBM MQ Console in einen I-Frame einzubetten.

```
https://*.example.com:*
```

Ermöglicht eine HTTPS-Webseite mit einem Hostnamen, der mit `example.com` endet, und die Verwendung eines beliebigen Ports, um die IBM MQ Console in einem I-Frame einzubetten.

Beispiel

Im folgenden Beispiel kann die IBM MQ Console in einen I-Frame von Webseiten eingebettet werden, die entweder von `https://site2.example.com:1234` oder `https://site2.example.com:1235` bereitgestellt werden:

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

CORS für die REST API konfigurieren

Standardmäßig sind im Web-Browser keine Scripts wie beispielsweise JavaScript für den Aufruf der REST API zulässig, wenn das Script nicht den gleichen Ursprung wie die REST API hat. Dies bedeutet, dass Kreuzursprungsanforderungen nicht aktiviert sind. Sie können Cross Origin Resource Sharing (CORS) konfigurieren, um Cross-Origin-Anforderungen von den angegebenen Ursprüngen zu ermöglichen.

Informationen zu diesem Vorgang

Sie können über einen Web-Browser auf die REST API zugreifen, beispielsweise über ein Script. Da diese Anforderungen eine andere Herkunft haben als die REST API, verweigert der Web-Browser die Anforderung, da es ich um eine Cross-Origin-Anforderung handelt. Der Ursprung ist unterschiedlich, wenn die Domäne, der Port oder das Schema nicht identisch ist.

Wenn Sie beispielsweise ein Script haben, das in `http://localhost:1999/` gehostet wird, stellen Sie eine Cross-origin-Anforderung ab, wenn Sie eine HTTP-Anforderung GET auf einer Website absetzen, die in `https://localhost:9443/` gehostet wird. Diese Anforderung ist eine Cross-Ursprungs-Anforderung, da die Portnummern und das Schema (HTTP) unterschiedlich sind.

Sie können Cross-Origin-Anforderungen aktivieren, indem Sie CORS konfigurieren und die Positionen für die Herkunft angeben, aus der auf die REST API zugegriffen werden kann.

Weitere Informationen zu Cross-Origin-Anforderungen finden Sie unter https://www.w3.org/TR/cors/ und https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

Vorgehensweise

1. Zeigen Sie die aktuelle Konfiguration an, indem Sie den folgenden Befehl eingeben:

   `dspmqweb properties -a`

   Der Eintrag `mqRestCorsAllowedOrigins` gibt den zulässigen Ursprung an. Der Eintrag `mqRestCorsMaxAgeInSeconds` gibt die Zeit in Sekunden an, in der der Web-Browser die Ergebnisse von CORS-Preflight-Prüfungen zwischenspeichern kann.
2. Geben Sie die Positionen für die Herkunft an, aus denen auf die REST API zugegriffen werden kann, indem Sie den folgenden Befehl eingeben:

   `setmqweb properties -k mqRestCorsAllowedOrigins -v` *`allowedOrigins`*

   Dabei gibt *allowedOrigins* den Ursprung an, von dem Sie Cross-Ursprungsanforderungen zulassen möchten. Sie können einen Stern (*) verwenden, der in Anführungszeichen ("*") eingeschlossen ist, um alle Kreuzursprungsanforderungen zu ermöglichen. Sie können mehr als einen Ursprung in eine durch Kommas getrennte Liste eingeben, die von doppelten Anführungszeichen umgeben ist. Um keine Cross-origin-Anforderungen zuzulassen, geben Sie als Wert für *allowedOrigins* leere Anführungszeichen ein.
3. Geben Sie die Zeit in Sekunden an, in der ein Web-Browser die Ergebnisse von CORS-Preflight-Prüfungen in den Cache stellen soll, indem Sie den folgenden Befehl eingeben:

   `setmqweb properties -k mqRestCorsMaxAgeInSeconds -v` *`time`*

Beispiel

Das folgende Beispiel zeigt die Cross-origin-Anforderungen, die für `http://localhost:9883`, `https://localhost:1999` und `https://localhost:9663` aktiviert sind. Das maximale Alter der zwischengespeicherten Ergebnisse von CORS-Preflight-Prüfungen wird auf 90 Sekunden gesetzt:

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://local
host:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

Validierung des Host-Headers für die IBM MQ Console und die REST API konfigurieren

Sie können den mqweb-Server so konfigurieren, dass der Zugriff auf die IBM MQ Console und die REST API in der Weise eingeschränkt wird, dass nur Anforderungen verarbeitet werden, die mit einem Host-Header gesendet werden, der mit einer angegebenen Zulassungsliste übereinstimmt. Bei Verwendung eines Host-Header-Wertes, der nicht auf der Zulassungsliste enthalten ist, wird ein Fehler zurückgegeben.

Informationen zu diesem Vorgang

Der mqweb-Server verwendet virtuelle Hosts, um die Zulassungsliste mit zulässigen Host-Headern zu definieren. Weitere Informationen zu virtuellen Hosts finden Sie in der WebSphere Liberty Dokumentation: https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html

Um diese Task auszuführen, müssen Sie ein Benutzer mit ausreichenden Berechtigungen sein, um die `mqwebuser.xml`-Datei zu bearbeiten:

- z/OS Unter z/OS müssen Sie Schreibzugriff auf die `mqwebuser.xml`-Datei haben.
- Multi Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

Vorgehensweise

1. Öffnen Sie die Datei `mqwebuser.xml`. Diese Datei befindet sich an einer der folgenden Positionen:
   - ALW

     Unter AIX, Linux, and Windows: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`
   - z/OS

     Unter z/OS: *WLP_user_directory*`/servers/mqweb`

     Dabei ist *WLP_Benutzerverzeichnis* das Verzeichnis, das angegeben wurde, als das Script **crtmqweb** ausgeführt wurde, um die mqweb-Serverdefinition zu erstellen.
2. Fügen Sie den folgenden Code in der `mqwebuser.xml`-Datei hinzu oder entfernen Sie das Kommentarzeichen:

   ```
   <virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
           <hostAlias>localhost:9080</hostAlias>
   </virtualHost>
   ```
3. Bearbeiten Sie das Feld **<hostAlias>** und fügen Sie die Kombination aus Hostname und Port ein, die Sie zulassen möchten.

   Dies kann die Kombination aus Hostname und Portname sein, die Sie bei der Konfiguration des mqweb-Servers verwendet haben. Wenn Sie beispielsweise die Standardkonfiguration von `localhost:9443`verwenden, können Sie `localhost:9443` im Feld **<hostAlias>** verwenden.

   Bei Bedarf können Sie mehrere **<hostAlias>** -Felder in den **<virtualHost>** -Tags hinzufügen, um mehr Kombinationen aus Hostname und Port zuzulassen. So können Sie beispielsweise Host-Header ermöglichen, die einen HTTP-Port verwenden, sowie Host-Header, die den HTTPS-Port verwenden.

Prüfprotokollierungs

Prüfsätze von Operationen, die in der IBM MQ Console und der REST API ausgeführt wurden, können durch das Aktivieren des Warteschlangenmanagerbefehls und von Konfigurationsereignissen aktiviert werden. Unter AIX, Linux, and Windows werden signifikante Statusänderungen in den Protokolldateien des mqweb-Servers aufgezeichnet.

Signifikante Statusänderungen

ALW

Unter AIX, Linux, and Windows erfasst die IBM MQ Console signifikante Statusänderungen als Nachrichten in den Protokollen des mqweb-Servers. Jede Nachricht gibt den Namen des authentifizierten Principals an, der die Operation angefordert hat.

Signifikante Statusänderungen, z. B. wenn Warteschlangenmanager erstellt, gestartet, beendet oder gelöscht werden, werden in den mqweb-Server `messages.log` -und `console.log` -Dateien auf der Protokollierungsstufe [AUDIT] protokolliert. Jeder Protokolleintrag gibt den Namen des authentifizierten Principals an, der die Operation angefordert hat.

Die Dateien `messages.log` und `console.log` befinden sich an der folgenden Position:

- ALW Unter AIX, Linux, and Windows: *MQ_DATA_PATH*`\web\installations\`*installationName*`\servers\mqweb\logs`

Weitere Informationen zum Konfigurieren der Ebenen für die Protokollierung des mqweb-Servers finden Sie unter Protokollierung konfigurieren.

Befehls-und Konfigurationsereignisse

Optional können Sie Befehls- und Konfigurationsereignisse im Warteschlangenmanager aktivieren, um Informationen zu den meisten IBM MQ Console- und REST API-Aktivitäten bereitzustellen. Die Erstellung von Kanälen und die Abfrage von Warteschlangen generieren z. B. Befehle und Konfigurationsereignisse. Weitere Informationen zum Aktivieren von Befehls-und Konfigurationsereignissen finden Sie im Abschnitt Steuerung von Konfigurations-, Befehls-und Protokollfunktionsereignissen .

Für diese Befehle und Konfigurationsereignisnachrichten wird das Feld MQIACF_EVENT_ORIGIN auf MQEVO_REST gesetzt, und das Feld MQCACF_EVENT_APPL_IDENTITY meldet die ersten 32 Zeichen des authentifizierten Principalnamens. Wenn ein Benutzer über die Rolle **MQWebAdmin** oder **MQWebAdminRO** verfügt, wird im Feld MQCACF_EVENT_USER_ID die Benutzer-ID des mqweb-Servers gemeldet und nicht der Benutzername des Prinzipals, der den Befehl ausgegeben hat. Wenn der Benutzer jedoch über die Rolle **MQWebUser** verfügt, meldet der MQCACF_EVENT_USER_ID den Benutzernamen des Principals, der den Befehl ausgegeben hat.

Zugehörige Konzepte

„Prüfprotokollierungs” auf Seite 510
Sie können mithilfe von Ereignisnachrichten auf Sicherheitseinbrüche oder unbefugte Zugriffe überprüfen. Sie können die Sicherheit Ihres Systems auch mit dem IBM MQ Explorer überprüfen.

Sicherheitsaspekte für die IBM MQ Console und die REST API on z/OS

Die Sicherheitsfunktionen von IBM MQ Console und REST API steuern, ob ein Benutzer Befehle ausgeben, anzeigen oder ändern kann. Die Befehle werden dann an den Warteschlangenmanager übergeben, und die WS-Manager-Sicherheit wird dann verwendet, um zu steuern, ob der Benutzer berechtigt ist, den Befehl an diesen bestimmten Warteschlangenmanager auszugeben.

Vorgehensweise

1. Stellen Sie sicher, dass die Benutzer-ID für die gestartete Task mqweb server die entsprechenden Berechtigungen zum Absetzen bestimmter PCF-Befehle und zum Zugriff auf bestimmte Warteschlan-

gen hat. Weitere Informationen finden Sie unter „Erforderliche Berechtigung für die Benutzer-ID der gestarteten Task 'mqweb server'" auf Seite 572.

2. Stellen Sie sicher, dass alle Benutzer, denen die Rolle `MQWebUser` erteilt wurde, über die entsprechenden Berechtigungen verfügen.

   Benutzer der IBM MQ Console und der REST API, die der Rolle `MQWebUser` zugeordnet sind, werden unter dem Sicherheitskontext des Principals ausgeführt. Diese Benutzer-IDs können nur Operationen ausführen, die der Benutzer-ID für die Ausführung auf dem Warteschlangenmanager erteilt wird. Sie müssen Zugriff auf dieselben Systemwarteschlangen wie den Adressraum des mqweb-Servers erhalten.

   Der Benutzer-ID der gestarteten MQweb-Server-Task muss ein alternativer Benutzerzugriff auf alle Benutzer erteilt werden, die der Rolle `MQWebUser` zugeordnet sind.

   Weitere Informationen zum Erteilen der entsprechenden Berechtigungen für Benutzer mit der `MQWebUser`-Rolle finden Sie in „Zugriff auf erforderliche IBM MQ-Ressourcen für die Verwendung der MQ Console oder REST API" auf Seite 573.

3. Optional: Konfigurieren Sie TLS für IBM MQ Console und REST API. Weitere Informationen finden Sie unter „TLS für REST API und IBM MQ Console auf z/OS konfigurieren" auf Seite 574.

z/OS Erforderliche Berechtigung für die Benutzer-ID der gestarteten Task 'mqweb server'

Unter z/OS sind für die Benutzer-ID der gestarteten Task für den mqweb-Server bestimmte Berechtigungen für die Ausgabe von PCF-Befehlen und den Zugriff auf Systemressourcen erforderlich.

Die Benutzer-ID für die Taskbenutzer-ID für den mqweb-Server benötigt

- Eine z/OS UNIX-Benutzer-ID (UID), um z/OS UNIX System Services verwenden zu können.
- Zugriff auf die `hlq.SCSQAUTH`- und `hlq.SCSQANL*`-Datensätze in der IBM MQ-Installation.
- Lesezugriff auf die IBM MQ-Installationsdateien in z/OS UNIX System Services.
- Lese- und Schreibzugriff auf das Liberty-Benutzerverzeichnis, das mit dem Script **crtmqweb** erstellt wurde.
- Berechtigung zum Herstellen einer Verbindung zum Warteschlangenmanageranager. Erteilen Sie dem mqweb-Server mit der Benutzer-ID *Lesen* den Zugriff auf das `hlq.BATCH`-Profil in der MQCONN-Klasse.
- Berechtigung zur Ausgabe von IBM MQ-Befehlen und Zugriff auf bestimmte Warteschlangen. Diese Einzelheiten werden in den Abschnitten „IBM MQ Console - erforderliche Profile für die Befehlssicherheit" auf Seite 246, „Sicherheit der Systemwarteschlange" auf Seite 222 und „Profile für Kontextsicherheit" auf Seite 233 beschrieben.
- Berechtigung zum Subskribieren des `SYSTEM.FTE` -Themas, um die REST API für MFTzu verwenden Erteilen Sie dem mqweb-Server mit der Benutzer-ID *ALTER* Zugriff auf das Profil `hlq.SUBSCRIBE.SYSTEM.FTE` in der Klasse MXTOPIC.
- Wenn Sie eine SAF-Registry konfigurieren, greifen Sie auf verschiedene Sicherheitsprofile zu. Weitere Informationen finden Sie unter „SAF-Registry für die IBM MQ Console und REST API konfigurieren" auf Seite 553.

Verbindungsauthentifizierung

Wenn Ihr Queue Manager so konfiguriert wurde, dass alle Stapelanwendungen eine gültige Benutzer-ID und ein gültiges Kennwort angeben, müssen Sie durch Setzen von CHKLOCL (REQUIRED) die Benutzer-ID *UPDATE* des mqweb-Servers mit dem Zugriff auf das `hlq.BATCH`-Profil in der MQCONN-Klasse aufrufen.

Diese Berechtigung bewirkt, dass die Verbindungsauthentifizierung im Modus CHKLOCL (OPTIONAL) für die Benutzer-ID der gestarteten Task 'mqweb server' ausgeführt wird.

Wenn Sie den Queue Manager nicht so konfiguriert haben, dass alle Stapelanwendungen eine gültige Benutzer-ID und ein gültiges Kennwort angeben müssen, reicht es aus, die Benutzer-ID anzugeben, die

den Zugriff auf das Profil `hlq.BATCH` in der MQCONN-Klasse mit dem Task *Lesen* des Typs 'mqweb server' startet.

Weitere Informationen zu CHCKLOCL finden Sie unter „CHCKLOCL für lokal gebundene Anwendungen verwenden” auf Seite 211.

Zugriff auf erforderliche IBM MQ-Ressourcen für die Verwendung der MQ Console oder REST API

Operationen, die in der MQ Console oder der REST API von einem Benutzer mit der Rolle `MQWebUser` ausgeführt werden, finden unter dem Sicherheitskontext des Benutzers statt.

Informationen zu diesem Vorgang

Im Abschnitt „Rollen in der IBM MQ Console und der REST API” auf Seite 555 finden Sie weitere Informationen zu den Rollen in der MQ Console und der REST API.

Gehen Sie folgendermaßen vor, um einem Benutzer mit der Rolle `MQWebUser` Zugriff auf die Warteschlangenmanagerressourcen zu erteilen, die für die Verwendung der MQ Console oder der REST API erforderlich sind.

Vorgehensweise

1. Erteilen Sie der `mqweb server started task` -Benutzer-ID alternativen Benutzerzugriff auf jede Benutzer-ID in der Rolle `MQWebUser` .

   Führen Sie dies auf jedem Warteschlangenmanager aus, den die Benutzer über die MQ Console oder die REST API verwalten.

   Sie können die folgenden RACF -Beispielbefehle verwenden, um der Benutzer-ID `mqweb server started task` alternativen Benutzerzugriff auf einen Benutzer in der Rolle `MQWebUser` zu erteilen:

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   Dabei gilt:

   `hlq`
   Ist das Profilpräfix, das entweder der Name des Warteschlangenmanagers oder der Name der Gruppe mit gemeinsamer Warteschlange sein kann.

   `userId`
   Ist der Benutzer in der Rolle `MQWebUser`

   `mqwebUserId`
   Ist die `mqweb server started task` -Benutzer-ID

   Anmerkung: Wenn Sie die Sicherheit in Groß-/Kleinschreibung verwenden, verwenden Sie die Klasse MXADMIN und nicht die Klasse MQADMIN.
2. Erteilen Sie jedem Benutzer im `MQWebUser`-Rollenzugriff Zugriff auf Systemwarteschlangen, die für die Verwendung von MQ Console und REST API erforderlich sind.

   Geben Sie dazu sowohl für SYSTEM.ADMIN.COMMAND.QUEUE als auch für SYSTEM.REST.REPLY.QUEUE jedem Benutzer UPDATE-Zugriff auf die Klassen MQQUEUE oder MXQUEUE an, je nachdem, ob die Groß-/Kleinschreibung in Groß-/Kleinschreibung verwendet werden soll.

   Sie müssen dies auf jedem Warteschlangenmanager tun, den der Benutzer über die REST APIverwalten wird, einschließlich der fernen Warteschlangenmanager, die über das administrative REST API-Gateway verwaltet werden.
3. Um einem Benutzer in der Rolle `MQWebUser` die Verwaltung ferner Warteschlangenmanager zu ermöglichen, erteilen Sie dem Benutzer UPDATE-Zugriff auf das Profil in der Klasse MQQUEUE oder MXQUEUE und schützen die Übertragungswarteschlange, die zum Senden von Befehlen an den fernen

Warteschlangenmanager verwendet wird. Beachten Sie, dass Sie dem Benutzer UPDATE Zugriff auf den Gateway-Warteschlangenmanager erteilen müssen.

Erteilen Sie auf dem fernen Warteschlangenmanager den Zugriff für denselben Benutzer in die Übertragungswarteschlange, die zum Senden von Befehlsantwortnachrichten an den Gateway-Warteschlangenmanager verwendet wird.

4. Erteilen Sie den Benutzern in der Rolle `MQWebUser` Zugriff auf alle anderen Ressourcen, die zur Ausführung der Operationen, die von der MQ Console und REST API unterstützt werden, erforderlich sind.

   Der Zugriff ist erforderlich für:

   - Das Ausführen von Operationen in der REST API wird in den Abschnitten zu den *Sicherheitsanforderungen* der jeweiligen REST API-Ressourcen beschrieben.
   - Die Ausgabe von Befehlen durch die MQ Console wird im Abschnitt „IBM MQ Console - erforderliche Profile für die Befehlssicherheit” auf Seite 246 beschrieben.

z/OS TLS für REST API und IBM MQ Console auf z/OS konfigurieren

Unter z/OS können Sie den mqweb-Server so konfigurieren, dass er einen RACF-Schlüsselring verwendet, um Zertifikate für sichere Verbindungen mit TLS und die Clientzertifikatsauthentifizierung zu speichern.

Vorbereitende Schritte

Sie müssen ein Benutzer sein, der über Schreibzugriff auf die Datei `mqwebuser.xml` und die Berechtigung zum Arbeiten mit SAF-Schlüsselringen verfügt, um dieses Verfahren abzuschließen.

Informationen zu diesem Vorgang

Die standardmäßige mqweb-Serverkonfiguration verwendet Java-Keystores für den Server und vertrauenswürdige Zertifikate. Unter z/OS können Sie den mqweb-Server so konfigurieren, dass er einen RACF-Schlüsselring anstatt der Java-Keystores verwendet. Der Server kann auch so konfiguriert werden, dass er Benutzern die Authentifizierung über ein Clientzertifikat ermöglicht.

Unter Liberty: Keystores erhalten Sie Informationen zur Verwendung von RACF-Schlüsselringen in Liberty.

Befolgen Sie diese Prozedur, um den mqweb-Server für die Verwendung eines RACF-Schlüsselrings und optional die Clientzertifikatsauthentifizierung zu konfigurieren. In diesem Verfahren werden die Schritte beschrieben, die zum Erstellen und Verwenden von Zertifikaten erforderlich sind, die mit Ihren eigenen Zertifikaten einer Zertifizierungsstelle (CA) signiert sind. Für die Produktion bevorzugen Sie möglicherweise Zertifikate, die von einer externen Zertifizierungsstelle angefordert werden.

Vorgehensweise

1. Erstellen Sie ein Zertifikat der Zertifizierungsstelle (Certificate Authority, CA), das zum Signieren des Serverzertifikats verwendet wird. Geben Sie zum Beispiel den folgenden RACF-Befehl ein:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb Certification Authority') -
    O('IBM') -
    OU('MQ')) -
  SIZE(2048) -
  WITHLABEL('mqwebCertauth')
```

2. Erstellen Sie ein Serverzertifikat, das mit dem in Schritt 1 erstellten CA-Zertifikat signiert wurde, indem Sie den folgenden Befehl eingeben:

```
RACDCERT ID(mqwebUserId) GENCERT -
  SUBJECTSDN(CN('hostname') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
```

```
   SIGNWITH (CERTAUTH LABEL('mqwebCertauth')) -
  WITHLABEL('mqwebServerCert')
```

Hierbei steht *mqweb-Benutzer-ID* für die Benutzer-ID der gestarteten Task 'mqweb server' und *Hostname* für den Hostnamen des mqweb-Servers.

3. Schließen Sie das CA-Zertifikat und das Serverzertifikat an einen SAF-Schlüsselring an, indem Sie die folgenden Befehle eingeben:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebCertauth') CERTAUTH)
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebServerCert'))
```

Hierbei steht *mqweb-Benutzer-ID* für die Benutzer-ID der gestarteten Task 'mqweb server' und *Schlüsselring* für den Namen des Schlüsselrings, den Sie verwenden möchten.

4. Exportieren Sie das CA-Zertifikat in eine CER-Datei, indem Sie den folgenden Befehl eingeben:

```
RACDCERT CERTAUTH EXPORT(LABEL('mqwebCertauth')) -
        DSN('hlq.CERT.MQWEBCA') -
        FORMAT(CERTDER) -
        PASSWORD('password')
```

5. Übertragen Sie das exportierte CA-Zertifikat per FTP auf Ihre Workstation und importieren Sie es als Zertifikat der Zertifizierungsstelle in Ihren Browser.
6. Optional: Wenn Sie die Clientzertifikatsauthentifizierung konfigurieren möchten, erstellen und exportieren Sie ein Clientzertifikat.
   a) Erstellen Sie ein Zertifikat der Zertifizierungsinstanz (CA), das zum Signieren des Clientzertifikats verwendet wird. Geben Sie zum Beispiel den folgenden RACF-Befehl ein:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb User CA') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
  WITHLABEL('mqwebUserCertauth')
```

   b) Schließen Sie das CA-Zertifikat an einen SAF-Schlüsselring an, indem Sie den folgenden Befehl eingeben:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebUserCertauth') CERTAUTH)
```

   Hierbei steht *mqweb-Benutzer-ID* für die Benutzer-ID der gestarteten Task 'mqweb server' und *Schlüsselring* für den Namen des Schlüsselrings, den Sie verwenden möchten.

   c) Erstellen Sie ein Clientzertifikat, das mit dem CA-Zertifikat signiert ist. Geben Sie zum Beispiel den folgenden Befehl ein:

```
RACDCERT ID(clientUserId) GENCERT -
  SUBJECTSDN(CN('clientUserId') -
     O('IBM') -
     OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebUserCertauth')) -
  WITHLABEL('userCertLabel')
```

   Hierbei steht *Client-Benutzer-ID* für den Benutzernamen.

   Die Methode, die zum Zuordnen eines Zertifikats zu einem Principal verwendet wird, hängt vom Typ der konfigurierten Benutzerregistry ab:

   - Wenn Sie eine Basisregistry verwenden, wird das Feld "Allgemeiner Name" im Zertifikat mit dem Benutzer in der Registry abgeglichen.
   - Wenn Sie eine SAF-Registry verwenden und sich das Zertifikat in der RACF-Datenbank befindet, wird der Zertifikatseigner verwendet, der beim Erstellen des Zertifikats mit dem Parameter **ID** angegeben wird.

- Wenn Sie eine LDAP-Registry verwenden, wird der vollständige definierte Name in dem Zertifikat mit der LDAP-Registry abgeglichen.

d) Exportieren Sie das Clientzertifikat in eine PKCS#12-Datei, indem Sie den folgenden Befehl eingeben:

```
RACDCERT ID(mqwebUserId) EXPORT(LABEL('userCertLabel')) -
         PASSWORD('password') DSN('hlq.USER.CERT')
```

e) Übertragen Sie das exportierte Zertifikat per FTP auf Ihre Workstation. Um das Clientzertifikat mit der IBM MQ Console zu verwenden, importieren Sie es in den Web-Browser, mit dem Sie auf die IBM MQ Console zugreifen, als persönliches Zertifikat.

7. Bearbeiten Sie die Datei *WLP_user_directory*/servers/mqweb/mqwebuser.xml, wobei *WLP_Benutzerverzeichnis* das Verzeichnis ist, das bei der Ausführung des Scripts **crtmqweb** angegeben wurde, um die mqweb-Serverdefinition zu erstellen.

   Nehmen Sie die folgenden Änderung vor, um den mqweb-Server für die Verwendung eines RACF-Schlüsselrings zu konfigurieren:

   a) Entfernen Sie die folgende Zeile oder kommentieren Sie sie aus:

   ```
   <sslDefault sslRef="mqDefaultSSLConfig"/>
   ```

   b) Fügen Sie die folgenden Anweisungen hinzu:

   ```
   <keyStore id="defaultKeyStore" filebased="false"
             location="safkeyring://mqwebUserId/keyring"
             password="password" readOnly="true" type="JCERACFKS" />
   <ssl id="thisSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLSv1.2"
             serverKeyAlias="mqwebServerCert" clientAuthenticationSupported="true" />
   <sslDefault sslRef="thisSSLConfig"/>
   ```

   Dabei gilt:

   - *mqweb-Benutzer-ID* ist die Benutzer-ID der gestarteten Task 'mqweb server'.
   - *Schlüsselring* ist der Name des RACF-Schlüsselrings.
   - *mqweb-Serverzertifikat* ist die Bezeichnung des mqweb-Serverzertifikats.

   Anmerkungen: Der Wert von **keyStore password** wird ignoriert.

8. Starten Sie den mqweb-Server erneut, indem Sie die gestartete Task des mqweb-Servers stoppen und erneut starten.
9. Optional: Verwenden Sie das Clientzertifikat zur Authentifizierung:

   - Wenn Sie das Clientzertifikat mit der IBM MQ Consoleverwenden möchten, geben Sie die URL für die MQ Console in den Web-Browser ein, in dem Sie das Clientzertifikat installiert haben.
   - Wenn Sie das Clientzertifikat mit der REST-API verwenden möchten, stellen Sie das Clientzertifikat mit jeder REST-Anforderung bereit.

   Anmerkungen:

   a. Wenn Sie nur Zertifikate für die Authentifizierung auf der IBM MQ Console verwenden, wird im Browser möglicherweise eine Liste der Zertifikate angezeigt, aus der Sie eine Auswahl treffen können.

   b. Wenn Sie ein anderes Zertifikat verwenden möchten, müssen Sie möglicherweise Ihren Browser schließen und erneut starten.

   c. Wenn Sie Clientzertifikate verwenden, die nicht in der RACF-Datenbank enthalten sind, können Sie RACF-Zertifikatsnamensfilter verwenden, um die Zertifikatsattribute einer Benutzer-ID zuzuordnen. Beispiel:

   ```
   RACDCERT ID(DEPT3USR) MAP SDNFILTER(OU=DEPT1.C=US)
   ```

   ordnet Zertifikate mit einem registrierten Namen zu, der OU=DEPT1 und C=US der Benutzer-ID DEPT3USR enthält.

Ergebnisse

Sie haben eine TLS-Schnittstelle für die IBM MQ Console und die REST API eingerichtet.

ALW Schlüssel und Zertifikate unter AIX, Linux, and Windows verwalten

Verwenden Sie unter AIX, Linux, and Windows die Befehle **runmqckm** und **runmqakm**, um Schlüssel, Zertifikate und Zertifikatanforderungen zu verwalten.

Der Befehl **runmqckm** stellt Funktionen bereit, die denen von **iKeyman**ähneln, und der Befehl **runmqakm** stellt Funktionen bereit, die denen von **gskitcapicmd**ähneln. Stellen Sie vor Verwendung von **runmqckm** oder **runmqakm**sicher, dass die Systemumgebungsvariablen ordnungsgemäß konfiguriert sind, indem Sie den Befehl **setmqenv** ausführen.

Für die Ausführung des Befehls **runmqckm** muss die IBM MQ-JRE-Komponente installiert sein. Falls diese Komponente nicht installiert ist, können Sie stattdessen den Befehl **runmqakm** ausführen.

Wenn Sie TLS-Zertifikate auf FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** anstelle des Befehls **runmqckm**. Dies liegt daran, dass der **runmqakm** -Befehl eine stärkere Verschlüsselung unterstützt.

Verwenden Sie die Befehle **runmqckm** und **runmqakm** , um Folgendes auszuführen:

- Typ der für IBM MQ erforderlichen CMS-Schlüsseldatenbankdateien erstellen
- Zertifikatsanforderungen erstellen
- Persönliche Zertifikate importieren
- CA-Zertifikate importieren
- Selbst signierte Zertifikate verwalten

Zugehörige Informationen

Keytool

ALW runmqckm-und runmqakm-Befehle unter AIX, Linux, and Windows

In diesem Abschnitt werden die Befehle **runmqckm** und **runmqakm** anhand der Befehlsobjekte beschrieben.

Die Hauptunterschiede zwischen den beiden Befehlen lauten wie folgt:

- **runmqckm**
  - Bietet Funktionen, die denen von **iKeycmd** ähneln
  - Unterstützt die Schlüssel-Repository-Dateiformate „JKS“ und „JCEKS“
- **runmqakm**
  - Bietet Funktionen, die denen von **gskitcapicmd** ähneln
  - Unterstützt die Erstellung von Zertifikaten und Zertifikatanforderungen mit öffentlichen Elliptic-Curve-Schlüsseln, anders als der Befehl **runmqckm**
  - Unterstützt eine stärkere Verschlüsselung der Schlüssel-Repository-Datei als der Befehl **runmqckm** mittels des Parameters **-strong**
  - Wurde als FIPS 140-2-konform zertifiziert und kann mit dem Parameter **-fips** für eine FIPS-konforme Verwendung konfiguriert werden

Achtung: Für den Befehl **runmqckm** ist die Installation der Funktion IBM MQ Java runtime environment (JRE) erforderlich.

Jeder Befehl gibt mindestens ein *Objekt* an. Befehle für PKCS#11-Einheitenoperationen können zusätzliche Objekte angeben. Befehle für Schlüsseldatenbank-, Zertifikats-und Zertifikatanforderungsobjekte geben auch eine *Aktion* an. Das Objekt kann eine der folgenden sein:

-keydb
Aktionen gelten für eine Schlüsseldatenbank

-cert
Aktionen gelten für ein Zertifikat

-certreq
Aktionen gelten für eine Zertifikatsanforderung

-help
Zeigt Hilfe an

-version
Zeigt Versionsinformationen an

In den folgenden Unterabschnitten werden die Aktionen beschrieben, die Sie in der Schlüsseldatenbank, dem Zertifikat und in Zertifikatsanforderungsobjekten ausführen können. Eine Beschreibung der Optionen für diese Befehle finden Sie im Abschnitt „runmqckm-und runmqakm-Optionen unter AIX, Linux, and Windows” auf Seite 591.

ALW Befehle für eine CMS-Schlüsseldatenbank nur unter AIX, Linux, and Windows

Sie können die Befehle **runmqckm** und **runmqakm** verwenden, um Schlüssel und Zertifikate für eine CMS-Schlüsseldatenbank zu verwalten.

-keydb -changepw
Ändern Sie das Kennwort für eine CMS-Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-keydb -changepw -db filename -pw password -new_pw new_password
-stash
```

Mit dem Befehl **runmqakm** :

```
-keydb -changepw -db filename -pw password -new_pw new_password
-stash -fips -strong
```

-keydb -create
Erstellen Sie eine CMS-Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-keydb -create -db filename -pw password -type cms -expire days
-stash
```

Mit dem Befehl **runmqakm** :

```
-keydb -create -db filename -pw password -type cms -expire days
-stash -fips -strong
```

-keydb -stashpw
Kennwort für eine CMS-Schlüsseldatenbank in eine Datei speichern:

Mit dem Befehl **runmqckm** :

```
-keydb -stashpw -db filename -pw password
```

Mit dem Befehl **runmqakm** :

```
-keydb -stashpw -db filename -pw password -fips
```

-cert -getdefault

Anmerkung: Das Standardzertifikat wird von IBM MQ 8.0 nicht unterstützt. Es sollte eine Zertifikatsbezeichnungskonfiguration gemäß der Beschreibung im Abschnitt „Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen” auf Seite 29 verwendet werden.

Rufen Sie das persönliche Standardzertifikat ab:

Mit dem Befehl **runmqckm** :

```
-cert -getdefault -db filename -pw password
```

Mit dem Befehl **runmqakm** :

```
-cert -getdefault -db filename -pw password -fips
```

-cert-ändern

Ändern Sie ein Zertifikat.

Anmerkung: Derzeit ist das einzige Feld, das geändert werden kann, das Feld "Certificate Trust".

Mit dem Befehl **runmqckm** :

```
-cert -modify -db filename -pw password -label label
-trust enable|disable
```

Mit dem Befehl **runmqakm** :

```
-cert -modify -db filename -pw password -label label
-trust enable|disable -fips
```

-cert -setdefault

Anmerkung: Das Standardzertifikat wird von IBM MQ 8.0 oder höher nicht unterstützt. Es sollte eine Zertifikatsbezeichnungskonfiguration gemäß der Beschreibung im Abschnitt „Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen” auf Seite 29 verwendet werden.

ALW Befehle für CMS-oder PKCS #12 -Schlüsseldatenbanken unter AIX, Linux, and Windows

Verwenden Sie die Befehle **runmqckm** und **runmqakm**, um Schlüssel und Zertifikate für eine CMS-Schlüsseldatenbank oder eine PKCS #12-Schlüsseldatenbank zu verwalten.

Anmerkung: IBM MQ unterstützt keine SHA-3- oder SHA-5-Algorithmen. Sie können die Namen der digitalen Signaturalgorithmen SHA384WithRSA und SHA512WithRSA verwenden, da beide Algorithmen zu Mitgliedern der SHA-2-Familie gehören.

Die Namen der digitalen Signaturalgorithmen SHA3WithRSA und SHA5WithRSA werden nicht weiter unterstützt, da sie eine abgekürzte Form von SHA384WithRSA bzw. SHA512WithRSA sind.

-keydb -changepw

Ändern Sie das Kennwort für eine Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-keydb -changepw -db filename -pw password -new_pw new_password -expire days
```

Mit dem Befehl **runmqakm** :

```
-keydb -changepw -db filename -pw password -new_pw new_password -expire days
-fips -strong
```

-keydb -convert

Wenn Sie den Befehl **runmqckm** verwenden, konvertieren Sie die Schlüsseldatenbank von einem Format ins andere:

```
-keydb -convert -db filename -pw password
-old_format cms | pkcs12 -new_format cms
```

Wenn Sie den Befehl **runmqakm** verwenden, konvertieren Sie eine CMS-Schlüsseldatenbank von einer alten Version in die neue Version:

```
-keydb -convert -db filename -pw password
-new_db filename -new_pw password -strong -fips
```

-keydb -create

Erstellen Sie eine Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-keydb -create -db filename -pw password -type cms
| pkcs12
```

Mit dem Befehl **runmqakm** :

```
-keydb -create -db filename -pw password -type cms
-fips -strong
```

-keydb -delete

Löschen Sie eine Schlüsseldatenbank:

Verwenden Sie einen der folgenden Befehle:

```
-keydb -delete -db filename -pw password
```

-keydb -list

Auflisten der derzeit unterstützten Typen von Schlüsseldatenbanken:

Mit dem Befehl **runmqckm** :

```
-keydb -list
```

Mit dem Befehl **runmqakm** :

```
-keydb -list -fips
```

-cert -add

Fügen Sie ein Zertifikat aus einer Datei in eine Schlüsseldatenbank hinzu:

Mit dem Befehl **runmqckm** :

```
-cert -add -db filename -pw password -label label -file filename
-format ascii | binary
```

Mit dem Befehl **runmqakm** :

```
-cert -add -db filename -pw password -label label -file filename
-format ascii | binary -fips
```

-cert -create

Erstellen Sie ein selbst signiertes Zertifikat:

Mit dem Befehl **runmqckm** :

```
-cert -create -db filename -pw password -label label
-dn distinguished_name -size 1024 | 512 -x509version 3 | 1 | 2
-expire days -sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
```

```
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

Mit dem Befehl **runmqakm** :

```
-cert -create -db filename -pw password -label label
-dn distinguished_name -size 2048 | 1024 | 512 -x509version 3 | 1 | 2
-expire days -fips -sig_alg md5 |
MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RSA | sha1 |
SHA1WithDSA | SHA1WithECDSA |
SHA1WithRSA | sha224 |
SHA224_WITH_RSA | SHA224WithDSA |
SHA224WithECDSA | SHA224WithRSA |
sha256 | SHA256_WITH_RSA |
SHA256WithDSA | SHA256WithECDSA |
SHA256WithRSA | SHA2WithRSA |
sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-cert -delete

Löschen Sie ein Zertifikat:

Mit dem Befehl **runmqckm** :

```
-cert -delete -db filename -pw password -label label
```

Mit dem Befehl **runmqakm** :

```
-cert -delete -db filename -pw password -label label -fips
```

-cert -details

Auflisten der detaillierten Informationen für ein bestimmtes Zertifikat:

Mit dem Befehl **runmqckm** :

```
-cert -details -db filename -pw password -label label
```

Mit dem Befehl **runmqakm** :

```
-cert -details -db filename -pw password -label label -fips
```

-cert -export

Exportieren Sie ein persönliches Zertifikat und den zugehörigen privaten Schlüssel aus einer Schlüsseldatenbank in eine PKCS#12-Datei oder in eine andere Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-cert -export -db filename -pw password -label label -type cms | pkcs12
-target filename -target_pw password -target_type  cms | pkcs12
```

Mit dem Befehl **runmqakm** :

```
-cert -export -db filename -pw password -label label -type cms | pkcs12
-target filename -target_pw password -target_type  cms | pkcs12
-encryption strong | weak -fips
```

-cert -extract

Extrahieren Sie ein Zertifikat aus einer Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-cert -extract -db filename -pw password -label label -target filename
-format ascii | binary
```

Mit dem Befehl **runmqakm** :

```
-cert -extract -db filename -pw password -label label -target filename
-format ascii | binary -fips
```

-cert -import
Importieren Sie ein persönliches Zertifikat aus einer Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label  label
```

Mit dem Befehl **runmqakm** :

```
-cert -import -file filename -pw password -type cms -target filename
-target_pw password -target_type cms -label  label -fips
```

Hinsichtlich beider Befehle:

- Die Option `-label` ist erforderlich und gibt die Bezeichnung für das Zertifikat an, das aus der Quellenschlüsseldatenbank importiert werden soll.
- Sie können zusätzlich die Option `-new_label` verwenden. Damit können importierte Zertifikate in der Zielschlüsseldatenbank einen anderen Kennsatz als in der ursprünglichen Schlüsseldatenbank erhalten.

-cert -list
Listet alle Zertifikate in einer Schlüsseldatenbank auf:

Mit dem Befehl **runmqckm** :

```
-cert -list all | personal | CA -db filename -pw password
```

Mit dem Befehl **runmqakm** :

```
-cert -list all | personal | CA -db filename -pw password -fips
```

-cert -receive
Empfangen Sie ein Zertifikat aus einer Datei:

Mit dem Befehl **runmqckm** :

```
-cert -receive -file filename -db filename -pw password
-format ascii | binary -default_cert yes | no
```

Mit dem Befehl **runmqakm** :

```
-cert -receive -file filename -db filename -pw password
-format ascii | binary -default_cert yes | no -fips
```

-cert -sign
Signieren Sie ein Zertifikat:

Mit dem Befehl **runmqckm** :

```
-cert -sign -db filename -file filename -pw password
-label label -target filename -format ascii | binary -expire days
-sig_alg MD2_WITH_RSA | MD2WithRSA | MD5_WITH_RSA |
MD5WithRSA | SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
```

```
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

Mit dem Befehl **runmqakm** :

```
-cert -sign -db filename -file filename -pw password
-label label -target filename -format ascii | binary -expire days -fips
-sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RSA | sha1 | SHA1WithDSA |
SHA1WithECDSA | SHA1WithRSA | sha224 |
SHA224_WITH_RSA | SHA224WithDSA |
SHA224WithECDSA | SHA224WithRSA | sha256 |
SHA256_WITH_RSA | SHA256WithDSA |
SHA256WithECDSA | SHA256WithRSA |
SHA2WithRSA | sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-certreq -create

Erstellen Sie eine Zertifikatsanforderung:

Mit dem Befehl **runmqckm** :

```
-certreq -create -db filename -pw password -label label -dn distinguished_name
-size 1024 | 512 -file filename
-sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

Mit dem Befehl **runmqakm** :

```
-certreq -create -db filename -pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -file filename -fips
-sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RSA | sha1 | SHA1WithDSA |
SHA1WithECDSA | SHA1WithRSA | sha224 |
SHA224_WITH_RSA | SHA224WithDSA |
SHA224WithECDSA | SHA224WithRSA | sha256 |
SHA256_WITH_RSA | SHA256WithDSA |
SHA256WithECDSA | SHA256WithRSA |
SHA2WithRSA | sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-certreq -delete

Löschen Sie eine Zertifikatsanforderung:

Mit dem Befehl **runmqckm** :

```
-certreq -delete -db filename -pw password -label  label
```

Mit dem Befehl **runmqakm** :

```
-certreq -delete -db filename -pw password -label  label -fips
```

-certreq -details
Auflisten der detaillierten Informationen zu einer bestimmten Zertifikatsanforderung:

Mit dem Befehl **runmqckm** :

```
-certreq -details -db filename -pw password -label  label
```

Mit dem Befehl **runmqakm** :

```
-certreq -details -db filename -pw password -label  label -fips
```

Listet die detaillierten Informationen zu einer Zertifikatsanforderung auf und zeigt die vollständige Zertifikatsanforderung an:

Mit dem Befehl **runmqckm** :

```
-certreq -details -showOID -db filename -pw password -label label
```

Mit dem Befehl **runmqakm** :

```
-certreq -details -showOID -db filename -pw password -label label -fips
```

-certreq -extract
Extrahieren Sie eine Zertifikatsanforderung aus einer Zertifikatsanforderungsdatenbank in eine Datei:

Für den Befehl **runmqckm** :

```
-certreq -extract -db filename -pw password -label label -target filename
```

Mit dem Befehl **runmqakm** :

```
-certreq -extract -db filename -pw password -label label -target filename -fips
```

-certreq -list
Listet alle Zertifikatsanforderungen in der Zertifikatsanforderungsdatenbank auf:

Mit dem Befehl **runmqckm** :

```
-certreq -list -db filename -pw password
```

Mit dem Befehl **runmqakm** :

```
-certreq -list -db filename -pw password -fips
```

-certreq -recreate
Erstellen Sie eine Zertifikatsanforderung erneut:

Mit dem Befehl **runmqckm** :

```
-certreq -recreate -db filename -pw password -label label -target filename
```

Mit dem Befehl **runmqakm** :

```
-certreq -recreate -db filename -pw password -label label -target filename -fips
```

ALW Befehle für Operationen mit Verschlüsselungseinheiten unter AIX, Linux, and Windows

Sie können die Befehle **runmqckm** (iKeycmd) und **runmqakm** verwenden, um Schlüssel und Zertifikate für Operationen mit Verschlüsselungseinheiten zu verwalten.

Anmerkung: IBM MQ unterstützt keine SHA-3- oder SHA-5-Algorithmen. Sie können die Namen der digitalen Signaturalgorithmen SHA384WithRSA und SHA512WithRSA verwenden, da beide Algorithmen zu Mitgliedern der SHA-2-Familie gehören.

Die Namen der digitalen Signaturalgorithmen SHA3WithRSA und SHA5WithRSA werden nicht weiter unterstützt, da sie eine abgekürzte Form von SHA384WithRSA bzw. SHA512WithRSA sind.

-keydb -changepw
Ändern Sie das Kennwort für eine Verschlüsselungseinheit:

Mit dem Befehl **runmqckm** :

```
-keydb -changepw -crypto module_name -tokenlabel token_label
-pw password -new_pw new_password
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-keydb -changepw -db filename -crypto module_name -tokenlabel  token_label
-pw password -new_pw new_password -fips -strong
```

-keydb -list
Auflisten der derzeit unterstützten Typen von Schlüsseldatenbanken:

Mit dem Befehl **runmqckm** :

```
-keydb -list
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-keydb -list -fips
```

-cert -add
Fügen Sie ein Zertifikat aus einer Datei zu einer Verschlüsselungseinheit hinzu:

Mit dem Befehl **runmqckm** :

```
-cert -add -crypto module_name -tokenlabel token_label -pw password
-label label -file filename -format ascii | binary
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -add -crypto module_name -tokenlabel token_label -pw password
-label label -file filename -format ascii | binary -fips
```

-cert -create
Erstellen Sie ein selbst signiertes Zertifikat auf einer Verschlüsselungseinheit:

Mit dem Befehl **runmqckm** :

```
-cert -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 1024 | 512 -x509version 3 | 1 | 2
-default_cert no | yes -expire days
-sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -x509version 3 | 1 | 2
-default_cert no | yes -expire days
-fips -sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RSA | sha1 | SHA1WithDSA |
SHA1WithECDSA | SHA1WithRSA |
sha224 | SHA224_WITH_RSA |
SHA224WithDSA | SHA224WithECDSA |
SHA224WithRSA | sha256 |
SHA256_WITH_RSA | SHA256WithDSA |
SHA256WithECDSA | SHA256WithRSA |
SHA2WithRSA | sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-cert -delete
Löschen Sie ein Zertifikat auf einer Verschlüsselungseinheit:

Mit dem Befehl **runmqckm** :

```
-cert -delete -crypto module_name -tokenlabel token_label -pw password -label label
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -delete -crypto module_name -tokenlabel token_label -pw password -label label -fips
```

-cert -details
Detaillierte Informationen zu einem bestimmten Zertifikat auf einer Verschlüsselungseinheit auflisten:

Mit dem Befehl **runmqckm** :

```
-cert -details -crypto module_name -tokenlabel token_label
-pw password -label label
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -details -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

Listen Sie die detaillierten Informationen auf und zeigen Sie das vollständige Zertifikat für ein bestimmtes Zertifikat auf einer Verschlüsselungseinheit an:

Mit dem Befehl **runmqckm** :

```
-cert -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label -fips
```

-cert -extract
Extrahieren Sie ein Zertifikat aus einer Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-cert -extract -crypto module_name -tokenlabel token_label -pw password
-label label -target filename -format ascii | binary
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -extract -crypto module_name -tokenlabel token_label -pw password
-label label -target filename -format ascii | binary -fips
```

-cert -import
Importieren Sie ein Zertifikat mit einer sekundären Schlüsseldatenbankunterstützung in eine Verschlüsselungseinheit:

Mit dem Befehl **runmqckm** :

```
-cert -import -db filename -pw password -label label -type cms
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -import -db filename -pw password -label label -type cms
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password -fips
```

Importieren Sie ein PKCS#12-Zertifikat in eine Verschlüsselungseinheit mit Unterstützung der sekundären Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-cert -import -file filename -pw password -type pkcs12
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -import -file filename -pw password -type pkcs12
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password -fips
```

-cert -list

Alle Zertifikate auf einer Verschlüsselungseinheit auflisten:

Mit dem Befehl **runmqckm** :

```
-cert -list all | personal | CA -crypto module_name
-tokenlabel token_label -pw  password
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -list all | personal | CA -crypto module_name
-tokenlabel token_label -pw  password -fips
```

-cert -receive

Empfangen Sie ein Zertifikat von einer Datei an eine Verschlüsselungseinheit mit Unterstützung der sekundären Schlüsseldatenbank:

Mit dem Befehl **runmqckm** :

```
-cert -receive -file filename -crypto module_name -tokenlabel  token_label
-pw password -default_cert yes | no -secondaryDB filename
-secondaryDBpw password -format  ascii | binary
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-cert -receive -file filename -crypto module_name -tokenlabel  token_label
-pw password -default_cert yes | no -secondaryDB filename
-secondaryDBpw password -format  ascii | binary -fips
```

-certreq -create
Erstellen Sie eine Zertifikatsanforderung auf einer Verschlüsselungseinheit:

Mit dem Befehl **runmqckm** :

```
-certreq -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 1024 | 512 -file filename
-sig_alg MD2_WITH_RSA | MD2WithRSA | MD5_WITH_RSA |
MD5WithRSA | SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-certreq -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -file filename -fips
-sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RA | sha1 | SHA1WithDSA |
SHA1WithECDSA | SHA1WithRSA |
sha224 | SHA224_WITH_RSA | SHA224WithDSA |
SHA224WithECDSA | SHA224WithRSA |
sha256 | SHA256_WITH_RSA | SHA256WithDSA |
SHA256WithECDSA | SHA256WithRSA |
SHA2WithRSA | sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-certreq -delete
Löschen Sie eine Zertifikatsanforderung von einer Verschlüsselungseinheit:

Mit dem Befehl **runmqckm** :

```
-certreq -delete -crypto module_name -tokenlabel token_label
-pw password -label label
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-certreq -delete -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

-certreq -details

Auflisten der detaillierten Informationen zu einer bestimmten Zertifikatsanforderung auf einer Verschlüsselungseinheit:

Mit dem Befehl **runmqckm** :

```
-certreq -details -crypto module_name -tokenlabel token_label
-pw password -label label
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-certreq -details -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

Listet die detaillierten Informationen zu einer Zertifikatsanforderung auf und zeigt die vollständige Zertifikatsanforderung auf einer Verschlüsselungseinheit an:

Mit dem Befehl **runmqckm** :

```
-certreq -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-certreq -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label -fips
```

-certreq -extract

Extrahieren Sie eine Zertifikatsanforderung aus einer Zertifikatsanforderungsdatenbank auf einer Verschlüsselungseinheit in eine Datei:

Mit dem Befehl **runmqckm** :

```
-certreq -extract -crypto module_name -tokenlabel token_label
-pw password -label label -target filename
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-certreq -extract -crypto module_name -tokenlabel token_label
-pw password -label label -target filename -fips
```

-certreq -list
Listet alle Zertifikatsanforderungen in der Zertifikatsanforderungsdatenbank auf einer Verschlüsselungseinheit auf:

Mit dem Befehl **runmqckm** :

```
-certreq -list -crypto module_name -tokenlabel token_label
-pw password
```

Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS#11-Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** 64-Bit-Programme sind. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Prozess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Einzige Ausnahmen sind die 32-Bit-Plattformen von Windows und Linux x86, da es sich bei den Programmen **strmqikm** und **runmqckm** auf diesen Plattformen um 32-Bit-Programme handelt.

Mit dem Befehl **runmqakm** :

```
-certreq -list -crypto module_name -tokenlabel token_label
-pw password -fips
```

ALW runmqckm-und runmqakm-Optionen unter AIX, Linux, and Windows

Sie können die Befehlszeilenoptionen **runmqckm** und **runmqakm** verwenden, um Schlüssel, Zertifikate und Zertifikatsanforderungen zu verwalten. **runmqckm** stellt Funktionen bereit, die denen von **iKeycmd** ähneln, und **runmqakm** stellt Funktionen bereit, die denen von **gskitcapicmd** ähneln.

Anmerkung: IBM MQ unterstützt keine SHA-3- oder SHA-5-Algorithmen. Sie können die Namen der digitalen Signaturalgorithmen SHA384WithRSA und SHA512WithRSA verwenden, da beide Algorithmen zu Mitgliedern der SHA-2-Familie gehören.

Die Namen der digitalen Signaturalgorithmen SHA3WithRSA und SHA5WithRSA werden nicht weiter unterstützt, da sie eine abgekürzte Form von SHA384WithRSA bzw. SHA512WithRSA sind.

Die Bedeutung einer Option kann von dem Objekt und der Aktion abhängen, die im Befehl angegeben sind.

*Tabelle 94. Optionen, die mit **runmqckm** und **runmqakm** verwendet werden können*

Parameter	Beschreibung
`-create`	Option zum Erstellen einer Schlüsseldatenbank.

*Tabelle 94. Optionen, die mit **runmqckm** und **runmqakm** verwendet werden können (Forts.)*	
Parameter	**Beschreibung**
`-crypto`	Der Name des Moduls für die Verwaltung einer PKCS#11-Verschlüsselungseinheit. Der Wert nach **-crypto** ist optional, wenn Sie den Modulnamen in der Eigenschaftendatei angeben. Wenn Sie Zertifikate oder Schlüssel verwenden, die auf PKCS #11 -Verschlüsselungshardware gespeichert sind, beachten Sie, dass **runmqckm** und **strmqikm** mit der Java Virtual Machine (JVM) ausgeführt werden, die mit der IBM MQ -Installation bereitgestellt wird. Das bedeutet, dass externe Module, die für die Unterstützung von PKCS #11 benötigt werden, in den JVM-Prozess geladen werden. Deshalb muss für die Verwaltung der Verschlüsselungshardware eine PKCS #11-Bibliothek installiert werden, die mit der Bitunterstützung der JVM übereinstimmt, und Sie müssen diese Bibliothek für die Verwendung von **runmqckm** oder **strmqikm** angeben.
-db	Vollständig qualifizierter Pfadname einer Schlüsseldatenbank.
-default_cert	Legt ein Zertifikat als Standardzertifikat fest. Der Wert kann `Ja` oder `Nein` sein. Der Standardwert ist `no`.
-dn	Definierter X.500-Name. Der Wert ist eine Zeichenfolge, die in doppelte Anführungszeichen eingeschlossen ist, z. B. `"CN=John Smith,O=IBM,OU=Test,C=GB"`. Beachten Sie, dass nur die Attribute O und C erforderlich sind. Die Angabe eines allgemeinen Namens (Common Name, CN) ist optional.
-encryption	Die Stärke der Verschlüsselung, die im Befehl zum Exportieren des Zertifikats verwendet wird. Der Wert kann `stark` oder `schwach` sein. Der Standardwert ist `strong` .
-expire	Verfallszeit in Tagen entweder eines Zertifikats oder eines Datenbankkennworts. Der Standardwert ist 365 Tage für ein Zertifikatkennwort. Es ist keine Standardzeit für ein Datenbankkennwort vorhanden: Verwenden Sie den Parameter **-expire** , um eine Verfallszeit für das Datenbankkennwort explizit festzulegen.
-file	Dateiname eines Zertifikats oder einer Zertifikatsanforderung.
-fips	Gibt an, dass der Befehl im FIPS-Modus ausgeführt wird. Im FIPS-Modus verwendet die ICC-Komponente Algorithmen, die FIPS 140-2 geprüft wurden. Wenn die ICC-Komponente nicht im FIPS-Modus initialisiert wird, schlägt der Befehl **runmqakm** fehl.
-format	Format eines Zertifikats. Der Wert kann `ascii` für Base64_encoded ASCII oder `binary` für binäre DER-Daten sein. Der Standardwert ist `ascii` .
-label	Kennsatz, der einer Zertifikats-oder Zertifikatsanforderung zugeordnet ist. Wenn es sich bei dem Zertifikat um ein persönliches Zertifikat handelt, das zum Identifizieren einer IBM MQ-Clientanwendung oder eines Warteschlangenmanagers verwendet wird, muss die Bezeichnung der IBM MQ-Zertifikatsbezeichnung (CERTLABL) entsprechen. Weitere Informationen finden Sie in „Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen” auf Seite 29.
-new_format	Neues Format der Schlüsseldatenbank.

*Tabelle 94. Optionen, die mit **runmqckm** und **runmqakm** verwendet werden können (Forts.)*	
Parameter	**Beschreibung**
-new_label	Wird in einem Zertifikatsimportbefehl verwendet, ermöglicht diese Option, dass ein Zertifikat mit einem anderen Kennsatz aus dem Kennsatz in der Quellenschlüsseldatenbank importiert werden kann. Wenn es sich bei dem Zertifikat um ein persönliches Zertifikat handelt, das zum Identifizieren einer IBM MQ-Clientanwendung oder eines Warteschlangenmanagers verwendet wird, muss die Bezeichnung der IBM MQ-Zertifikatsbezeichnung (CERTLABL) entsprechen. Weitere Informationen finden Sie in „Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen” auf Seite 29.
-new_pw	Neues Datenbankkennwort.
-old_format	Altes Format der Schlüsseldatenbank.
-pw	Kennwort für die Schlüsseldatenbank oder PKCS#12-Datei.
-secondaryDB	Der Name einer sekundären Schlüsseldatenbank für PKCS#11-Einheitenoperationen.
-secondaryDBpw	Kennwort für die sekundäre Schlüsseldatenbank für PKCS#11-Einheitenoperationen.
-showOID	Zeigt das vollständige Zertifikat oder die Zertifikatsanforderung an.
-sig_alg	Der Hash-Algorithmus, der bei der Erstellung einer Zertifikatsanforderung, eines selbst signierten Zertifikats oder der Unterzeichnung eines Zertifikats verwendet wird. Dieser Hashing-Algorithmus wird verwendet, um die Signatur zu erstellen, die dem neu erstellten Zertifikat oder der Zertifikatsanforderung zugeordnet ist. Folgende Werte sind für **runmqckm** möglich: MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256_WITH_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA, SHAWithRSA. Der Standardwert ist SHA1WithRSA . Für **runmqakm** sind folgende Werte möglich: md5, MD5_WITH_RSA, MD5WithRSA, SHA_WITH_DSA, SHA_WITH_RSA, sha1, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, sha224, SHA224_WITH_RSA, SHA224WithDSA, SHA224WithECDSA, SHA224WithRSA, sha256, SHA256_WITH_RSA, SHA256WithDSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithRSA, sha384, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, sha512, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHAWithDSA, SHAWithRSA, EC_ecdsa_with_SHA1, EC_ecdsa_with_SHA224, EC_ecdsa_with_SHA256, EC_ecdsa_with_SHA384 oder EC_ecdsa_with_SHA512. Der Standardwert ist SHA1WithRSA .

*Tabelle 94. Optionen, die mit **runmqckm** und **runmqakm** verwendet werden können (Forts.)*	
Parameter	**Beschreibung**
-size	Schlüsselgröße. Für **runmqckm**kann der Wert 512, 1024 oder 2048 sein. Der Standardwert ist 1024 Bit. Für **runmqakm**hängt der Wert vom Signaturalgorithmus ab: • Für RSA-Signaturalgorithmen (der Standardalgorithmus, der verwendet wird, wenn kein **-sig_alg** angegeben ist), kann der Wert 512, 1024, 2048 oder 4096 sein. Eine RSA-Schlüsselgröße von 512 Bit ist nicht zulässig, wenn der Parameter **-fips** aktiviert ist. Die standardmäßige RSA-Schlüsselgröße ist 2048 Bit. • Bei Elliptic Curve-Algorithmen kann der Wert 256, 384 oder 512 sein. Die standardmäßige Elliptic Curve-Schlüsselgröße hängt von dem Signaturalgorithmus ab. Für SHA256 ist es 256; für SHA384 ist es 384; und für SHA512 ist es 512.
-stash	Stoppt das Kennwort der Schlüsseldatenbank in einer Datei. Gilt nur für Datenbanken des Typs CMS und PKCS12. **Anmerkung: -stash** gilt für **-keydb** -create-Befehle, die **runmqckm**/**runmqakm** anweisen, eine Stashdatei mit dem Kennwort zu erstellen. Wenn Sie den Befehl `$ runmqakm -help` absetzen, werden nur die allgemeinen Hilfeparameter aufgelistet.
-stashed	Gibt an, dass sich das Kennwort für die Schlüsseldatenbank oder die PKCS #12 -Datei in einer Stashdatei befindet. **Anmerkung:** Die Option **-stashed** ist bei Aufrufen außer den **-keydb** -create-Befehlen gültig. Wenn Sie diese Option nicht angeben, müssen Sie das Kennwort mit **-pw**angeben. Außerdem wird die ausführliche Hilfe **-stashed** nur angezeigt, wenn Sie dem Befehl mitteilen, welche Art von Aktion Sie ausführen.
-target	Zieldatei oder Datenbank.
-target_pw	Kennwort für die Schlüsseldatenbank, wenn **-target** eine Schlüsseldatenbank angibt.
-target_type	Der Typ der durch den Operanden **-target** angegebenen Datenbank. Der Parameter **-type** enthält die zulässigen Werte.
-tokenLabel	Kennsatz einer PKCS#11-Verschlüsselungseinheit.
-trust	Trust-Status eines CA-Zertifikats. Der Wert kann `enable` oder `disable` sein. Der Standardwert ist `enable` .
-type	Typ der Datenbank. Folgende Werte sind möglich: • `cms` für eine CMS-Schlüsseldatenbank • `pkcs12` für eine PKCS#12-Datei.
-x509version	Version des zu erstellenden X.509-Zertifikats. Der Wert kann 1, 2 oder 3 sein. Der Standardwert ist 3.

*Tabelle 94. Optionen, die mit **runmqckm** und **runmqakm** verwendet werden können (Forts.)*	
Parameter	**Beschreibung**

-rfc3339	Verwenden Sie diesen Parameter, um das Datum im RFC 3339-Format für den Befehl `runmqakm -cert -details` auszugeben, der das folgende Format hat:
	`Not Before : 2015-08-26T08:53:37Z`
	`Not After : 2016-08-26T08:53:37Z`
	Beachten Sie, dass der Parameter **-rfc3339** nach den zusätzlichen Parametern in dem Befehl angezeigt werden muss:
	`runmqakm -cert -details -db exampleDB -stashed -label certficateLabel -rfc3339`

ALW runmqakm-Fehlercodes unter AIX, Linux, and Windows

Eine Tabelle mit den von **runmqakm**ausgegebenen numerischen Fehlercodes und deren Bedeutung.

Fehlercode	Fehlernachricht
0	Erfolg
1	Unbekannter Fehler
2	Fehler bei Ver-/Entschlüsselung ASN.1.
3	Fehler beim Initialisieren des Programms für Ver-/Entschlüsselung ASN.1.
4	ASN.1-Ver-/Entschlüsselungsfehler, da Index außerhalb des zulässigen Bereichs oder Wahlfeld nicht vorhanden.
5	Ein Datenbankfehler ist aufgetreten.
6	Fehler beim Öffnen der Datenbankdatei. Prüfen Sie, ob die Datei vorhanden ist und Berechtigungen vorliegen.
7	Fehler beim erneuten Öffnen der Datenbankdatei.
8	Datenbankerstellung fehlgeschlagen.
9	Die Datenbank ist bereits vorhanden.
10	Fehler beim Löschen der Datenbankdatei.
11	Datenbank konnte nicht geöffnet werden.
12	Fehler beim Lesen der Datenbankdatei.
13	Fehler beim Schreiben von Daten in die Datenbankdatei.
14	Gültigkeitsfehler bei der Datenbank.
15	Ungültige Datenbankversion festgestellt.
16	Ungültiges Datenbankkennwort festgestellt.
17	Ungültiger Datenbankdateityp festgestellt.
18	Die angegebene Datenbank ist beschädigt.

Fehlercode	Fehlernachricht
19.	Ein ungültiges Kennwort wurde bereitgestellt oder die Schlüsseldatenbank wurde manipuliert oder beschädigt.
20	Integritätsfehler in Datenbankschlüsseleintrag.
21	Die Datenbank enthält bereits ein identisches Zertifikat.
22	In database(Record ID) ist bereits ein identischer Schlüssel vorhanden.
23	Ein Zertifikat mit derselben Bezeichnung ist bereits in der Schlüsseldatenbank vorhanden.
24	In database(Signature) ist bereits ein identischer Schlüssel vorhanden.
25	In database(Unsigned Certificate) ist bereits ein identischer Schlüssel vorhanden.
26	In database(Issuer and Serial Number) ist bereits ein identischer Schlüssel vorhanden.
27	In database(Subject Public Key Info) ist bereits ein identischer Schlüssel vorhanden.
28	In database(Unsigned CRL) ist bereits ein identischer Schlüssel vorhanden.
29	Der Kennsatz wurde bereits in der Datenbank verwendet.
30	Ein Kennwortverschlüsselungsfehler ist aufgetreten.
31	Ein LDAP-bezogener Fehler ist aufgetreten. (LDAP wird von diesem Programm nicht unterstützt)
32	Ein Verschlüsselungsfehler ist aufgetreten.
33	Ein Ver-/Entschlüsselungsfehler ist aufgetreten.
34	Ungültiger Verschlüsselungsalgorithmus festgestellt.
35	Fehler beim Unterzeichnen von Daten.
36	Fehler beim Prüfen von Daten.
273	Fehler beim Berechnen des Datenauszugs.
38	Ungültiger Verschlüsselungsparameter festgestellt.
39	Nicht unterstützter Verschlüsselungsalgorithmus festgestellt.
40	Die angegebene Eingabegröße überschreitet die unterstützte Modulusgröße.
41	Nicht unterstützte Modulusgröße festgestellt.
42	Gültigkeitsfehler bei der Datenbank.

Fehlercode	Fehlernachricht
43	Gültigkeitsprüfung für Schlüsseleintrag fehlgeschlagen.
44	Erweiterungsfeld doppelt vorhanden.
45	Falsche Schlüsselversion.
46	Ein erforderliches Erweiterungsfeld fehlt.
47	Der Gültigkeitszeitraum schließt das heutige Datum nicht ein oder fällt nicht in den Gültigkeitszeitraum des Ausstellers
48	Gültigkeitszeitraum enthält nicht den heutigen Tag oder liegt nicht im Gültigkeitszeitraum des Ausstellers.
49	Fehler beim Auswerten der Erweiterung für die Nutzung des privaten Schlüssels.
50	Aussteller des Schlüssels nicht gefunden.
51	Erforderliche Zertifikaterweiterung fehlt.
52	Ungültige Erweiterung für Basisvorgabe gefunden.
53	Gültigkeitsprüfung für Schlüsselunterschrift fehlgeschlagen.
54	Stammschlüssel des Schlüssels ist nicht gesichert.
55	Schlüssel wurde widerrufen.
56	Fehler beim Auswerten der Erweiterung des Tags des Berechtigungsschlüssels.
57	Fehler beim Auswerten der Erweiterung für die Nutzung des privaten Schlüssels.
58	Fehler beim Auswerten der Erweiterung für den Alternativnamen des Subjekts.
59	Fehler beim Auswerten der Erweiterung für den Alternativnamen des Ausstellers.
60	Fehler beim Auswerten der Erweiterung für die Nutzung des Schlüssels.
61	Unbekannte kritische Erweiterung festgestellt.
62	Fehler beim Auswerten von Schlüsselpaareinträgen.
63	Fehler bei der CRL-Gültigkeitsprüfung.
64	Es ist ein Mutex-Fehler aufgetreten.
65	Ungültiger Parameter festgestellt.
66	Ein Nullparameter oder ein Speicherzuordnungsfehler wurde festgestellt.
67	Anzahl oder Größe zu groß oder zu klein.
68	Das alte Kennwort ist ungültig.

Fehlercode	Fehlernachricht
69	Das neue Kennwort ist ungültig.
70	Das Kennwort ist abgelaufen.
71	Ein Thread-Fehler ist aufgetreten.
72	Fehler beim Erstellen von Threads.
73	Während ein Thread auf die Beendigung wartete, ist ein Fehler aufgetreten.
74	Ein E/A-Fehler ist aufgetreten.
75	Fehler beim Laden von CMS.
76	Ein Fehler im Zusammenhang mit der Verschlüsselungshardware ist aufgetreten.
77	Die Routine für die Initialisierung der Bibliothek konnte nicht erfolgreich aufgerufen werden.
78	Die interne Kennungstabelle der Datenbank ist beschädigt.
79	Es ist ein Speicherzuordnungsfehler aufgetreten.
80	Nicht erkannte Option festgestellt.
81	Fehler beim Abrufen der Zeitinformation.
82	Es ist ein Mutex-Erstellungsfehler aufgetreten.
83	Fehler beim Öffnen des Nachrichtenkatalogs.
84	Fehler beim Öffnen des Fehlernachrichtenkatalogs.
85	Name einer Leerdatei festgestellt.
86	Fehler beim Öffnen von Dateien. Prüfen Sie, ob die Datei vorhanden ist und Berechtigungen vorliegen.
87	Fehler beim Öffnen zu lesender Dateien.
88	Fehler beim Öffnen zu schreibender Dateien.
89	Diese Datei ist nicht vorhanden.
90	Datei kann aufgrund ihrer Berechtigungseinstellung nicht geöffnet werden.
91	Fehler beim Schreiben von Daten in Dateien.
92	Fehler beim Löschen von Dateien.
93	Ungültige Daten mit Base64-Verschlüsselung festgestellt.
94	Ungültige Base64-Nachrichtenart festgestellt.
95	Fehler beim Verschlüsseln von Daten nach Base64-Verschlüsselungsregel.
96	Fehler beim Entschlüsseln von Daten mit Base64-Verschlüsselung.
97	Fehler beim Abruf einer registrierten Namenskennung.

Fehlercode	Fehlernachricht
98	Feld für erforderlichen allgemeinen Namen ist leer.
99	Feld für erforderlichen Landes- oder Regionsnamen ist leer.
100	Ungültige Datenbankkennung festgestellt.
101	Die Schlüsseldatenbank ist nicht vorhanden.
102	Datenbank mit den Anforderungsschlüsselpaaren nicht vorhanden.
103	Die Kennwortdatei ist nicht vorhanden.
104	Das neue Kennwort ist mit dem alten identisch.
105	Kein Schlüssel in der Schlüsseldatenbank gefunden.
106	Kein Anforderungsschlüssel gefunden.
107	Keine verlässliche CA gefunden.
108	Kein Anforderungsschlüssel für das Zertifikat gefunden.
109	Kein Standardschlüssel in Schlüsseldatenbank.
110	Kein Standardschlüssel in der Schlüsseldatenbank.
111	Kein Standardschlüssel im Schlüsseldatensatz.
112	Kein Zertifikat im Schlüsseldatensatz.
113	Kein CRL-Eintrag vorhanden.
114	Ungültiger Dateiname für Schlüsseldatenbank festgestellt.
115	Nicht erkannte private Schlüsselart.
116	Eingabe eines ungültigen registrierten Namens wurde festgestellt.
117	Kein Schlüsseleintrag mit dem angegebenen Schlüsselkennsatz gefunden.
118	Die Schlüsselkennsatzliste wurde beschädigt.
119	Eingabedaten sind keine gültigen PKCS12-Daten.
120	Das Kennwort ist ungültig oder die PKCS12-Daten sind fehlerhaft bzw. wurden mit einer neueren PKCS12-Version erstellt
121	Nicht erkannte Schlüsselexportart festgestellt.
122	Nicht unterstützter, auf Kennwort basierender Verschlüsselungsalgorithmus wurde festgestellt.
123	Fehler beim Umwandeln der Schlüsselringdatei in eine CMS-Schlüsseldatenbank.
124	Fehler beim Umwandeln der CMS-Schlüsseldatenbank in eine Schlüsseldatei.

Fehlercode	Fehlernachricht
125	Fehler beim Erstellen eines Zertifikats für die Zertifikatanforderung.
126	Erstellung einer vollständigen Ausstellerkette nicht möglich.
127	Ungültige WEBDB-Daten festgestellt.
128	Keine Daten zum Schreiben in die Schlüsselringdatei vorhanden.
129	Ablaufdatum des Zertifikats liegt zu weit in der Zukunft.
130	Das Kennwort ist zu kurz; es muss aus mindestens {0} Zeichen bestehen.
131	Ein Kennwort muss mindestens ein numerisches Zeichen enthalten.
132	Alle Zeichen im Kennwort sind entweder alphabetische oder numerische Zeichen.
133	Es wurde ein nicht erkannter oder nicht unterstützter Signaturalgorithmus angegeben.
134	Ungültiger Datenbanktyp festgestellt.
135	Die angegebene sekundäre Schlüsseldatenbank wird durch andere PKCS#11-Einheit verwendet.
136	Keine sekundäre Schlüsseldatenbank angegeben.
137	Kennsatz ist auf der PKCS#11-Einheit nicht vorhanden.
138	Kennwort für den Zugriff auf die PKCS#11-Einheit erforderlich.
139	Kennwort für den Zugriff auf die PKCS#11-Einheit nicht erforderlich.
140	Verschlüsselungsbibliothek kann nicht geladen werden.
141	PKCS#11 wird für diese Operation nicht unterstützt.
142	An der PKCS#11-Einheit ist eine Operation fehlgeschlagen.
143	Der LDAP-Benutzer ist kein gültiger Benutzer. (LDAP wird von diesem Programm nicht unterstützt)
144	Der LDAP-Benutzer ist kein gültiger Benutzer. (LDAP wird von diesem Programm nicht unterstützt)
145	LDAP-Abfrage fehlgeschlagen. (LDAP wird von diesem Programm nicht unterstützt)
146	Ungültige Zertifikatskette festgestellt.

Fehlercode	Fehlernachricht
147	Stammzertifikat nicht als vertrauenswürdig anerkannt.
148	Widerrufenes Zertifikat festgestellt.
149	Funktion für Verschlüsselungsobjekt fehlgeschlagen.
150	Es ist keine Datenquelle für die Zertifikatswiderrufsliste verfügbar.
151	Kein Verschlüsselungstoken verfügbar.
152	FIPS-Modus nicht verfügbar.
153	Es liegt ein Konflikt mit den Einstellungen des FIPS-Modus vor.
154	Eingegebenes Kennwort erfüllt die Mindestanforderungen an die Kennwortsicherheit nicht.
200	Während der Initialisierung des Programms ist ein Fehler aufgetreten.
201	Zerlegung in Tokens der Argumente wird an das fehlgeschlagene Programm 'runmqakm' übermittelt.
202	Das im Befehl angegebene Objekt ist kein erkanntes Objekt.
203	Die Aktion ist keine bekannte -keydb-Aktion.
204	Die Aktion ist keine bekannte -cert-Aktion.
205	Die Aktion ist keine bekannte -certreq-Aktion.
206	Für den angeforderten Befehl fehlt ein Tag.
207	Der Wert, der mit dem Tag -version übergeben wurde, ist kein erkannter Wert.
208	Der Wert, der mit dem Tag -size übergeben wurde, ist kein erkannter Wert.
209	Der Wert, der mit dem Tag -dn übergeben wurde, hat kein korrektes Format.
210	Der Wert, der mit dem Tag -format übergeben wurde, ist kein erkannter Wert.
211	Dem Öffnen der Datei ist ein Fehler zugeordnet.
212	PKCS12 wird in diesem Stadium nicht unterstützt.
213	Der Verschlüsselungstoken, für den Sie das Kennwort ändern möchten, ist nicht kennwortgeschützt.
214	PKCS12 wird in diesem Stadium nicht unterstützt.
215	Eingegebenes Kennwort erfüllt die Mindestanforderungen an die Kennwortsicherheit nicht.
216	FIPS-Modus nicht verfügbar.

Fehlercode	Fehlernachricht
217	Die von Ihnen als Ablaufdatum eingegebene Anzahl an Tagen liegt außerhalb des zulässigen Bereichs.
218	Die Kennwortsicherheit hat nicht die Mindestanforderungen erfüllt.
219	Kein Standardzertifikat in der angeforderten Schlüsseldatenbank gefunden.
220	Ungültiger Anerkennungsstatus festgestellt.
221	Nicht unterstützter Signaturalgorithmus festgestellt. In diesem Stadium werden nur MD5 und SHA1 unterstützt.
222	PCKS11 wird für diese bestimmte Operation nicht unterstützt.
223	Die übergebene Aktion ist keine bekannte -random-Aktion.
224	Eine Feldlänge kleiner als Null ist nicht zulässig.
225	Wenn der Tag -strong verwendet wird, beträgt die Mindestlänge des Kennworts 14 Zeichen.
226	Wenn der Tag -strong verwendet wird, beträgt die Maximallänge des Kennworts 300 Zeichen.
227	Der MD5-Algorithmus wird im FIPS-Modus nicht unterstützt.
228	Der site-Tag wird für den Befehl -cert -list wird nicht unterstützt. Dieses Attribut wird für Abwärtskompatibilität und potenzielle zukünftige Erweiterungen hinzugefügt.
229	Der Wert, der dem Tag -ca zugeordnet ist, wird nicht erkannt. Der Wert muss 'true' oder 'false' sein.
230	Der Wert, der mit dem Tag -type übergegeben wurde, ist ungültig.
231	Der Wert, der mit dem Tag -expire übergeben wurde, liegt unter dem zulässigen Bereich.
232	Der verwendete oder angeforderte Verschlüsselungsalgorithmus wird nicht unterstützt.
233	Das Ziel ist bereits vorhanden.

V 9.2.0 V 9.2.0 Kennwörter in den Konfigurationsdateien der IBM MQ-Komponente schützen

Um bestimmte Funktionen von IBM MQ verwenden zu können, müssen Kennwörter möglicherweise direkt in IBM MQ oder in Konfigurationsdateien, die von diesem Feature gelesen werden, bereitgestellt werden. Ab IBM MQ 9.2.0 wird ein neues Kennwortschutzsystem implementiert, das den Schutz von Kennwörtern in diesen Konfigurationsdateien ermöglicht.

Sie sollten Kennwörter in Konfigurationsdateien schützen. In der folgenden Liste wird die allgemeine Terminologie erläutert, die für jede Komponente verwendet wird:

Ursprünglicher Schlüssel
Der Verschlüsselungsschlüssel, den Sie für die Verwendung im Verschlüsselungsprozess bereitstellen.

Für jede der aufgeführten Komponenten können Sie eine Datei mit dem ursprünglichen Schlüssel bereitstellen, in der sich der zu verwendende Verschlüsselungsschlüssel befindet, wenn die in der Konfigurationsdatei dieser Komponente gespeicherten Kennwörter geschützt oder gelesen werden.

Die Datei muss eine einzelne Zeile mit mindestens einem Zeichen enthalten.

Es gibt keine Begrenzung oder Voraussetzung für die Länge des Verschlüsselungsschlüssels. Ihre Schlüsseldatei sollte aber mindestens 16 Zeichen enthalten. Ihre Datei kann beispielsweise Folgendes enthalten:

```
Th1sIs@n3Ncypt|onK$y
```

Zusätzlich sollte in der von Ihnen bereitgestellten Datei mit dem ursprünglichen Schlüssel Folgendes berücksichtigt werden:

- Sie sollte einen eindeutigen Verschlüsselungsschlüssel enthalten
- Sie sollte mit den Betriebssystemberechtigungen auseichend geschützt sein.

Ursprünglicher Standardschlüssel
Dies ist der verwendete Standardverschlüsselungsschlüssel, wenn Sie beim Verschlüsseln von Daten keinen ursprünglichen Schüssel bereitstellen. Sie sollten den ursprünglichen Standardschlüssel allerdings **nicht** verwenden.

Einfache Textzeichenfolge
Die Zeichenfolge, die verschlüsselt, gewöhnlich ein Kennwort

Verschlüsseltes Kennwort
Eine Zeichenfolge mit dem verschlüsselten Kennwort in einem Format, das von IBM MQ verstanden wird.

Wichtig: Verschlüselte Kennwortzeichenfolgen, die Sie für die Verwendung mit einer Komponente generiert haben, können nicht in die Konfigurationsdatei zur Verwendung mit einer anderen Komponente kopiert werden. Jedes Kennwort in jeder Komponente muss mit dem für die Komponente spezifischen Dienstprogramm geschützt sein.

Einzelheiten zum Schützen von Kennwörter für jede Komponente von IBM MQ, die den Kennwortschutz unterstützt, sind in den folgenden Abschnitten aufgeführt:

- Advanced Message Security
- „Managed File Transfer” auf Seite 604
- „IBM MQ Internet Pass-Thru” auf Seite 605
- Deprecated „IBM MQ Bridge to blockchain” auf Seite 606
- Deprecated „IBM MQ Bridge to Salesforce” auf Seite 607
- V 9.2.3 „IBM MQ-Clients mit Verschlüsselungshardware” auf Seite 607

Advanced Message Security

Für Advanced Message Security (AMS) Java-Clients ist der Zugriff auf einen Keystore mit privaten Schlüsseln erforderlich, damit eine Nachricht geschützt werden kann.

V 9.2.2 Advanced Message Security (AMS) MQI-Clients oder Queue Manager, die für die Ausführung von MCA-Interception konfiguriert sind, benötigen möglicherweise Zugriff auf PKCS#11-Verschlüsselungshardware oder PEM-Dateien, die private Schlüssel zum Schutz von Nachrichten enthalten.

Um auf diese, muss ein Passwort namens `keystore.conf` in der AMS-Konfigurationsdatei bereitgestellt werden. Verwenden Sie den Befehl **runamscred** , um die in der Datei `keystore.conf` enthaltenen sensiblen Informationen zu schützen. For example:

```
runamscred -f <keystore configuration file>
```

Der Befehl **runamscred** schützt sensible Parameter in der Datei, die mit dem Flag **-f** angegeben wird.

V 9.2.2 Der IBM MQ -Installation wurden zwei **runamscred** -Programme hinzugefügt:

- Ein MQI- **runamscred** -Programm, das sich in `<IBM MQ installation root>/bin` befindet
- Ein Java **runamscred** -Programm, das sich in `<IBM MQ installation root>/java/bin` befindet

Achtung:

1. V 9.2.2 Um die Kompatibilität sicherzustellen, verwenden Sie das Programm Java **runamscred**, um die Konfigurationsdateien zu schützen, die mit Java AMS-Clients und dem MQI-**runamscred**-Programm verwendet werden sollen, um Konfigurationsdateien zu schützen, die mit den MQI-AMS-Clients verwendet werden sollen.
2. Sie sollten sicherstellen, dass nach der Ausführung von **runamscred** alle sensiblen Informationen geschützt sind.
3. Sie können die geschützte Datei den für AMS aktivierten Anwendungen als normale Datei bereitstellen.

Wenn Sie die Datei mit dem ursprünglichen Schlüssel für die Verwendung während der Ausführung von AMS-Anwendungen oder beim Schützen einer Konfigrationsdatei mit dem Keystore mithilfe des Befehls **runamscred** überschreiben oder bereitstellen, verwenden Sie einen der folgenden vier Verfahren. Dies sind die folgenden Verfahren in der Reihenfolge ihrer Priorität:

1. Parameter **-sf** (nur **runamscred**)
2. Über die Umgebungsvariable MQS_AMSCRED_KEYFILE
3. Parameter **amscred.keyfile** in der Konfigurationsdatei
4. Datei mit dem ursprünglichen Standardschlüssel, wenn keine der oben genannten Optionen angegeben ist.

Achtung: V 9.2.2 Sie sollten den ursprünglichen Standardschlüssel nicht verwenden.

Vor IBM MQ 9.2 wurde ein anderes System für den Kennwortschutz in AMS Java-Konfigurationsdateien verwendet.

Das Programm **runamscred** verwendet standardmäßig das neue System für den Schutz von Kennwörtern. Das bedeutet, dass neue Konfigurationsdateien nicht mit älteren Versionen von AMS Java kompatibel sind. Wenn Sie Konfigurationsdateien mit dem alten System zum Kennwortschutz verwenden möchten, verwenden Sie das Flag **-sp** *0*.

Managed File Transfer

Managed File Transfer (MFT) speichert Berechtigungsnachweise, die für den Zugriff auf Warteschlangenmanager oder andere Ressourcen in mehreren XML-Eigenschaftendateien erforderlich sind:

- `MQMFTCredentials.xml`-Berechtigungsnachweise für die Verbindung zu Agenten-, Koordinations- und Befehls-Queue Managern und Kennwörtern für die Verbindung zu Keystores für die sichere Kommunikation.
- `ProtocolBridgeCredentials.xml`-Berechtigungsnachweise für die Verbindung zu Protokollservern, z. B. FTP/SFTP/FTPS.
- `ConnectDirectCredentials.xml`-Berechtigungsnachweise für den Connect:Direct-Agenten für die Verbindung zu einem Connect:Direct-Knoten.

Weitere Informationen finden Sie unter „Gespeicherte Berechtigungsnachweise in MFT verschlüsseln” auf Seite 609.

Zum Schutz sensibler Informationen, die in diesen Dateien gespeichert werden, verwenden Sie den Befehl fteObfuscate in der angegebenen Datei mit dem Flag **-f** . For example:

```
fteObfuscate -f <File to protect>
```

Wenn Sie eine Datei mit dem ursprünglichen Schlüssel bereitstellen möchten, der beim Schützen Ihrer MFT-Konfigurationen verwendet wird, verwenden Sie das Flag **-sf**:

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

Wenn Sie keinen ursprünglichen Schlüssel bereitstellen, wird ein Standardschlüssel zum Schützen der sensiblen Informationen verwendet, obwohl Sie diese Option nicht verwenden sollten.

Achtung:

1. Sie sollten sicherstellen, dass nach der Ausführung von **fteObfuscate** alle sensiblen Informationen geschützt sind.
2. Sie können die geschützte Datei als normale Datei für MFT bereitstellen.

Stellen Sie die Datei mit dem ursprünglichen Schlüssel während der Ausführung über die folgenden drei Verfahren bereit. Dies sind die folgenden Verfahren in der Reihenfolge ihrer Priorität:

1. Durch Verwendung einer Java -Systemeigenschaft.
   - V 9.2.0.15 Vor IBM MQ 9.2.0 Fix Pack 15wurde der Name dieser Java -Systemeigenschaft im Produktcode als `com.ibm.wqmfte.cred.keyfile`falsch geschrieben. Ab IBM MQ 9.2.0 Fix Pack 15lautet die Schreibweise des Eigenschaftsnamens `com.ibm.wmqfte.cred.keyfile`. Managed File Transfer verwendet beide Versionen der Systemeigenschaft Java , wenn geprüft wird, ob ein Benutzer eine Datei angegeben hat, die den ursprünglichen Schlüssel für die Verschlüsselung und Entschlüsselung von Berechtigungsnachweisen enthält. Dies ermöglicht die Verwendung der korrekten Schreibweise des Eigenschaftsnamens unter Wahrung der Kompatibilität mit einer früheren Version mit dem alten falsch geschriebenen Namen. Wenn beide Java -Systemeigenschaften festgelegt sind, wird der Wert der korrekt geschriebenen Eigenschaft `com.ibm.wmqfte.cred.keyfile` verwendet.
   - Verwenden Sie vor IBM MQ 9.2.0 Fix Pack 15die Eigenschaft `com.ibm.wqmfte.cred.keyfile`.
2. In den Eigenschaftendateien für Agenten, Protokollfunktionen, Befehlen und die Koordination.
3. In der `installation.properties`-Datei.

Vor IBM MQ 9.2 wurde zum Schutz von Berechtigungsnachweisen in den MFT-Konfigurationsdateien ein anderes System für den Schutz von Berechtigungsnachweisen verwendet.

Standardmäßig schützt **fteObfuscate** Berechtigungsnachweise mit dem neuen System. Das bedeutet, dass Konfigurationsdateien nicht mit älteren Versionen von MFT kompatibel sind.

Wenn Sie Konfigurationsdateien mit dem alten System zum Schutz von Berechtigungsnachweisen verwenden möchten, verwenden Sie das Flag **-sp***0*.

IBM MQ Internet Pass-Thru

Die Konfigurationsdatei IBM MQ Internet Pass-Thru (MQIPT) kann Kennwörter für den Zugriff auf verschiedene Ressourcen sowie für das MQIPT-Verwaltungskennwort enthalten.

Sie können diese Kennwörter mit dem Befehl **mqiptPW** schützen, der mit MQIPT bereitgestellt wird.

```
mqiptPW
```

Geben Sie für den Schutz eines Kennworts mit einem bestimmten ursprünglichen Schlüssel das Flag **-sf** an:

```
mqiptPW -sf <intial key file>
```

Weitere Informationen finden Sie im Abschnitt Kennwortverschlüsselungsschlüssel angeben.

Wenn Sie keinen ursprünglichen Schlüssel bereitstellen, wird ein Standardschlüssel zum Schützen der sensiblen Informationen verwendet, obwohl Sie diese Option nicht verwenden sollten.

mqiptPW fordert Sie zur sicheren Eingabe eines zu schützenden Kennworts ein und gibt eine Zeichenfolge zurück, die in die MQIPT-Konfigurationsdatei kopiert werden muss.

Stellen Sie die Datei mit dem ursprünglichen Schlüssel während der Ausführung über die folgenden vier Verfahren bereit. Dies sind die folgenden Verfahren in der Reihenfolge ihrer Priorität:

1. Über den Parameter **-sf** beim Starten von MQIPT.
2. In der Umgebungsvariablen MQS_MQIPTCRED_KEYFILE.
3. In der Eigenschaft **com.ibm.mq.ipt.cred.keyfile** Java .
4. In einer Datei mit dem Namen `mqipt_cred.key` im Ausgangsverzeichnis von MQIPT ist dies das Verzeichnis, in dem die Konfigurations- und Protokolldateien von MQIPT und andere enthalten sind.

Vor IBM MQ 9.2 wurde zum Schutz von Berechtigungsnachweisen in den MQIPT-Konfigurationsdateien ein anderes System für den Schutz von Berechtigungsnachweisen verwendet.

Standardmäßig schützt **mqiptPW** Berechtigungsnachweise, die das neue System verwenden. Dies bedeutet, dass Konfigurationsdateien nicht mit älteren Versionen von MQIPT kompatibel sind.

Wenn Sie Schlüsselspeicherkennwörter mit dem alten Berechtigungsnachweissystem schützen möchten, verwenden Sie die Befehlssyntax **mqiptPW**, die in früheren Versionen als IBM MQ 9.2 unterstützt wird.

IBM MQ Bridge to blockchain

Deprecated

Bridge to blockchain-Konfigurationen werden in Dateien gespeichert, die mit dem Befehl **runmqbcb** generiert werden können. Bei der Ausführung dieses Befehls werden Sie aufgefordert, die Kennwörter und eine Position mit der zu verwendenden Datei mit dem ursprünglichen Schlüssel sicher bereitzustellen.

Zum Überschreiben der Datei mit dem ursprünglichen Schlüssel, die während der Ausführung oder im Konfigurationsmodus verwendet wird, nutzen Sie das Flag **-sf**. So generieren Sie beispielsweise eine Konfiguration mit einer bestimmten Datei für den ursprünglichen Schlüssel:

```
runmqbcb -o <output file> -sf <initial key file>
```

Gehen Sie folgendermaßen vor, um während der Ausführung eine bestimmte Datei für den ursprünglichen Schlüssel zu verwenden:

```
runmqbcb -f <config file> -sf <initial key file>
```

Vor IBM MQ 9.2 wurde zum Schutz von Berechtigungsnachweisen in den Bridge to blockchain-Konfigurationsdateien ein anderes System für den Schutz von Berechtigungsnachweisen verwendet.

Standardmäßig schützt **runmqbcb** Berechtigungsnachweise mit dem neuen System. Das bedeutet, dass Konfigurationsdateien nicht mit älteren Versionen von Bridge to blockchain kompatibel sind.

Wenn Sie Konfigurationsdateien mit dem alten System zum Schutz von Berechtigungsnachweisen verwenden möchten, verwenden Sie das Flag **-sp***0*.

Wichtig:

- Deprecated IBM MQ Bridge to blockchain gilt in allen Releases ab 22. November 2022 als veraltet (siehe US-Ankündigungsschreiben 222-341).

- V 9.2.0.21 Removed Für Long Term Supportwird IBM MQ Bridge to blockchain in IBM MQ 9.2.0 CSU 21entfernt.

IBM MQ Bridge to Salesforce

Deprecated

Bridge to Salesforce-Konfigurationen werden in Dateien gespeichert, die mit dem Befehl **runmqsfb** generiert werden können. Bei der Ausführung dieses Befehls werden Sie aufgefordert, die Kennwörter und eine Position mit der zu verwendenden Datei mit dem ursprünglichen Schlüssel sicher bereitzustellen.

Zum Überschreiben der Datei mit dem ursprünglichen Schlüssel, die während der Ausführung oder im Konfigurationsmodus verwendet wird, nutzen Sie das Flag **-sf**. So generieren Sie beispielsweise eine Konfiguration mit einer bestimmten Datei für den ursprünglichen Schlüssel:

```
runmqsfb -o <output file> -sf <initial key file>
```

Gehen Sie folgendermaßen vor, um während der Ausführung eine bestimmte Datei für den ursprünglichen Schlüssel zu verwenden:

```
runmqsfb -f <config file> -sf <initial key file>
```

Vor IBM MQ 9.2 wurde zum Schutz von Berechtigungsnachweisen in den Bridge to Salesforce-Konfigurationsdateien ein anderes System für den Schutz von Berechtigungsnachweisen verwendet.

Standardmäßig schützt **runmqfsb** Berechtigungsnachweise mit dem neuen System. Das bedeutet, dass Konfigurationsdateien nicht mit älteren Versionen von Bridge to Salesforce kompatibel sind.

Wenn Sie Konfigurationsdateien mit dem alten System zum Schutz von Berechtigungsnachweisen verwenden möchten, verwenden Sie das Flag **-sp***0*.

Wichtig: IBM MQ Bridge to Salesforce gilt in allen Releases ab 22. November 2022 als veraltet (siehe US-Ankündigungsschreiben 222-341).

IBM MQ-Clients mit Verschlüsselungshardware

V 9.2.3

Sie können IBM MQ-Clients so konfigurieren, dass sie PKCS- #11-Verschlüsselungshardware verwenden, um private Schlüssel und Zertifikate zu speichern, die in der TLS-Kommunikation verwendet werden. Um auf PKCS- #11 -Einheiten zugreifen zu können, müssen Sie ein Kennwort als Teil der Konfigurationszeichenfolge angeben, die für IBM MQ client bereitgestellt wird.

Wichtig: Kennwörter, die über MQSC0bereitgestellt werden.Die **SSLCryptoHardware** -Strukturzeichenfolge oder das Warteschlangenmanagerattribut **SSLCRYP** kann mit diesem Verfahren nicht geschützt werden.

Sie können dieses Kennwort mit dem Befehl **runp11cred** schützen, der sich im Ordner 'bin' befindet, der im IBM MQ-Installationsstammverzeichnis gefunden wurde.

Der Befehl **runp11cred** fordert Sie zur sicheren Eingabe eines Kennworts auf, das geschützt werden soll, und gibt eine Zeichenfolge zurück, die in die Konfigurationszeichenfolge für die Verschlüsselungshardware kopiert werden muss.

Beispiel: Wenn Sie GSK_PKCS11 verwenden, gehen Sie wie folgt vor:

`GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;`**`Passw0rd`**`;SYMMETRIC_CIPHER_ON`

Geben Sie anschließend **Passw0rd** ein, wenn Sie dazu aufgefordert werden. **runp11cred** gibt eine Zeichenfolge zurück, der folgenden Zeichenfolge ähnelt:

<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==

Kopieren Sie die Zeichenfolge in Fettschrift anstelle von **Passw0rd** in die GSK_PKCS11-Zeichenfolge:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!0TyDxrRaS6JUs-
jON9zfK6S4wEHm SNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

Um ein Kennwort mit einem bestimmten Anfangsschlüssel zu schützen, verwenden Sie einen der folgenden Mechanismen. Dies sind die folgenden Verfahren in der Reihenfolge ihrer Priorität:

1. Parameter **-sf** (nur Befehl**runp11cred**)
2. Umgebungsvariable "MQS_SSLCRYP_KEYFILE"
3. **SSLCryptoHardwareKeyFile** SSL-Zeilengruppenattribut (nur IBM MQ client)
4. Datei mit dem ursprünglichen Standardschlüssel, wenn keine der oben genannten Optionen angegeben ist.

Achtung: Sie sollten den ursprünglichen Standardschlüssel nicht verwenden.

Schutz von Datenbankauthentifizierungsdetails

Wenn Sie den Benutzernamen und die Kennwortauthentifizierung verwenden, um eine Verbindung zum Datenbankmanager herzustellen, können Sie sie im MQ-XA-Berechtigungsnachweisspeicher speichern, um zu vermeiden, dass das Kennwort in Klartext in der `qm.ini`-Datei gespeichert wird.

XAOpenString für den Ressourcenmanager aktualisieren

Um den Berechtigungsnachweisspeicher zu verwenden, müssen Sie XAOpenString in der `qm.ini`-Datei ändern. Die Zeichenfolge wird verwendet, um eine Verbindung zum Datenbankmanager herzustellen. Sie geben ersetzbare Felder an, um anzugeben, wo der Benutzername und das Kennwort in der XAOpenString-Zeichenfolge ersetzt werden.

- Das Feld +USER+ wird durch den im XACredentials-Speicher gespeicherten Benutzernamenswert ersetzt.
- Das Feld +PASSWORD+ wird durch den Kennwortwert ersetzt, der im Geschäft "XACredentials" gespeichert ist.

Die folgenden Beispiele zeigen, wie eine XAOpenString-Datei geändert wird, um die Berechtigungsnachweisdatei für die Verbindung mit der Datenbank zu verwenden.

Verbindung zu einer Db2-Datenbank herstellen

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

Verbindung zu einer Oracle-Datenbank herstellen

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

Mit den Berechtigungsnachweisen für die Datenbank in den MQ XA-Berechtigungsnachweise-Speicher arbeiten

Nachdem Sie die Datei `qm.ini` mit den austauschbaren Berechtigungsnachweiszeichenfolgen aktualisiert haben, müssen Sie den Benutzernamen und das Kennwort mit dem Befehl **setmqxacred** zum MQ -Speicher für Berechtigungsnachweise hinzufügen. Sie können auch **setmqxacred** verwenden, um vorhandene Berechtigungsnachweise zu ändern, zu löschen oder aufzulisten. In den folgenden Beispielen werden einige typische Anwendungsfälle genannt:

Berechtigungsnachweise hinzufügen

Mit dem folgenden Befehl werden der Benutzername und das Kennwort für den Warteschlangenmanager QM1 für die Ressource mqdb2 sicher gespeichert.

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

Berechtigungsnachweise aktualisieren

Setzen Sie den Befehl **setmqxacred** mit dem neuen Benutzernamen und dem neuen Kennwort erneut ab, um den Benutzernamen und das Kennwort zu aktualisieren, die für die Verbindung zu einer Datenbank verwendet werden:

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

Sie müssen den WS-Manager erneut starten, damit die Änderungen wirksam werden.

Berechtigungsnachweise löschen

Mit dem folgenden Befehl werden die Berechtigungsnachweise gelöscht:

```
setmqxacred -m QM1 -x mydb2 -d
```

Berechtigungsnachweise auflisten

Mit dem folgenden Befehl werden Berechtigungsnachweise aufgelistet:

```
setmqxacred -m QM1 -l
```

Zugehörige Verweise

setmqxacred

Managed File Transfer sichern

Direkt nach der Installation, wenn noch keine Änderungen vorgenommen wurden, hat Managed File Transfer eine Sicherheitsstufe, die eventuell für Test- und Bewertungszwecke in einer geschützten Umgebung ausreicht. In einer Produktionsumgebung hingegen sollte kontrolliert werden, wer Dateiübertragungsoperationen starten kann und wer über Lese- und Schreibzugriff auf die übertragenen Dateien verfügt. Außerdem spielt hier der Schutz der Dateiintegrität eine Rolle.

Zugehörige Tasks

Einschränken von Gruppenberechtigungen für MFT-spezifische Ressourcen

Berechtigungen für MFT-spezifische Ressourcen verwalten

„Advanced Message Security mit Managed File Transfer verwenden” auf Seite 679

In diesem Szenario wird erläutert, wie Advanced Message Security konfiguriert wird, um die Vertraulichkeit von Nachrichten für Daten bereitzustellen, die über Managed File Transfer gesendet werden.

Zugehörige Verweise

Berechtigungen für MFT für den Zugriff auf Dateisysteme

MFT-Eigenschaft 'commandPath'

Berechtigung zur Veröffentlichung von Protokoll- und Statusnachrichten von MFT-Agenten

V 9.2.0 V 9.2.0 Gespeicherte Berechtigungsnachweise in MFT verschlüsseln

Managed File Transfer (MFT) erfordert mehrere Benutzer-IDs und Berechtigungsnachweise, die in zwei XML-Dateien gespeichert sind. Sie können diese mit dem Befehl **fteObfuscate** verschlüsseln. Ab IBM MQ 9.2.0bietet dieser Befehl einen erweiterten Schutz der gespeicherten Berechtigungsnachweise.

Berechtigungsnachweisdateien

`MQMFTCredentials.xml`
Diese Datei enthält die Benutzer-ID und die Berechtigungsnachweise, mit denen eine Verbindung zu Agenten und Koordinations- und Befehlswarteschlangenmanager hergestellt wird. Die Berechtigungsnachweise für den Zugriff auf Keystores für sichere Verbindungen zu Warteschlangenmanagern werden ebenfalls in dieser Datei gespeichert.

Details zu den Eigenschaftswerten, die die Position der `MQMFTCredentials.xml` -Datei definieren, finden Sie in „Verbindungsauthentifizierung für MFT und IBM MQ" auf Seite 613.

`ProtocolBridgeCredentials.xml`
Diese Datei enthält die Benutzer-ID und die Berechtigungsnachweise, mit denen eine Verbindung zu Protokollservern hergestellt wird.

Berechtigungsnachweise mit dem Befehl `fteObfuscate` verschlüsseln

Ab IBM MQ 9.2.0akzeptiert der Befehl **`fteObfuscate`** die folgenden Parameter:

- **`credentialsFileName`** (erforderlich)
- **`protection mode`**, **`credentialsKeyFile`** und **`outputFileName`** (alle optional)

Einzelheiten zu den Parametern finden Sie unter **`fteObfuscate`**.

Wenn Sie den Schutzmodus oder die Schlüsseldatei für Berechtigungsnachweise nicht angeben, verwendet der Befehl den Standardschutzmodus und verschlüsselt die Berechtigungsnachweise mit dem letzten Algorithmus, allerdings mit einem festen Schlüssel.

Wenn Sie den Schutzmodus `0`, aber keine Schlüsseldatei für Berechtigungsnachweise angeben, funktioniert der Befehl wie in früheren Releases des Produkts. Sie erhalten eine Warnung in der Konsole, in der auf die Verwendung eines veralteten Schutzes hingewiesen wird.

Wenn Sie den Schutzmodus `0` und eine Schlüsseldatei für Berechtigungsnachweise angeben, erhalten Sie in der Konsole eine Fehlernachricht, in der angezeigt wird, dass die Angabe einer Schlüsseldatei beim Verwenden von Schutzmodus `0` nicht zulässig ist.

Wenn Sie den Schutzmodus `1`, aber keine Schlüsseldatei für Berechtigungsnachweise angeben, verschlüsselt der Befehl die Berechtigungsnachweise mit dem letzten Algorithmus, allerdings mit einem festen Schlüssel.

Wenn Sie den Schutzmodus `1` und eine Schlüsseldatei für Berechtigungsnachweise angeben, verschlüsselt der Befehl die Berechtigungsnachweise mit dem letzten Algorithmus.

Wenn Sie den Schutzmodus `1` oder keinen Schutzmodus und eine Schlüsseldatei für Berechtigungsnachweise angeben, die nicht vorhanden ist, wird in der Konsole ein Fehler ausgegeben, in dem angezeigt wird, dass die Datei nicht vorhanden ist.

Wenn Sie den Schutzmodus `1` oder keinen Schutzmodus und eine Schlüsseldatei für Berechtigungsnachweise angeben, die nicht gelesen werden kann, wird in der Konsole ein Fehler ausgegeben, in dem angezeigt wird, dass die Datei nicht gelesen werden kann.

V 9.2.4 Wenn Sie den Schutzmodus 2angeben und keine Schlüsseldatei für Berechtigungsnachweise angeben, verwendet der Befehl den Schutzmodus 2, um Berechtigungsnachweise mit dem neuesten Algorithmus zu verschlüsseln, und einen festen Schlüssel, um sie zu verschlüsseln.

V 9.2.4 Wenn Sie den Zugriffsschutzmodus 2und eine Schlüsseldatei für Berechtigungsnachweise angeben, verwendet der Befehl den Zugriffsschutzmodus 2 zum Verschlüsseln von Berechtigungsnachweisen mit dem neuesten Algorithmus und einen benutzerdefinierten Schlüssel zum Verschlüsseln.

V 9.2.4 Wenn Sie den Schutzmodus 2 oder keinen Schutzmodus und eine Schlüsseldatei für Berechtigungsnachweise angeben, die nicht vorhanden ist, wird in der Konsole ein Fehler ausgegeben, in dem angezeigt wird, dass die Datei nicht vorhanden ist.

V 9.2.4 Wenn Sie den Schutzmodus 2 oder keinen Schutzmodus und eine Schlüsseldatei für Berechtigungsnachweise angeben, die nicht gelesen werden kann, wird in der Konsole ein Fehler ausgegeben, in dem angezeigt wird, dass die Datei nicht gelesen werden kann.

Berechtigungsnachweise entschlüsseln

Sie können den Pfad zur Datei mit dem ursprünglichen Schlüssel an verschiedenen Positionen angeben. Zur Entschlüsselung von Berechtigungsnachweisen, die mit einem anderen ursprünglichen Schlüssel als dem Standardschlüssel verschlüsselt wurden, muss der Name der Datei mit dem ursprünglichen Schlüssel für MFT auf eine der folgenden Arten bereitgestellt werden, die hier nach Priorität aufgelistet sind:

1. Verwenden der Java Virtual Machine (JVM)-Eigenschaft `com.ibm.wqmfte.cred.keyfile`, zum Beispiel:

   ```
   -Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
   ```

2. Durch Festlegen einer Eigenschaft in einer Eigenschaftendatei für einen Agenten, einen Befehl, eine Koordination oder eine Protokollfunktion. Der Name der Eigenschaftendatei und die Eigenschaft, die in ihr festgelegt werden muss, sind in der folgenden Tabelle aufgeführt:

Eigenschaftendatei	Eigenschaftsname
agent.properties	agentCredentialsKeyFile
command.properties	commandCredentialsKeyFile
coordination.properties	coordinationCredentialsKeyFile
logger.properties	loggerCredentialsKeyFile

3. In der Datei installation.properties.

   Anstatt Eigenschaften in einzelnen Eigenschaftendateien hinzuzufügen, können Sie die Eigenschaft **commonCredentialsKeyFile** zur vorhandenen allgemeinen Datei `installation.properties` hinzufügen, sodass Agent, Protokollfunktion und Befehle dieselbe Eigenschaft verwenden können.

Möglicherweise haben Sie verschiedene **CredentialsKeyFile**-Eigenschaften an mehreren Positionen definiert, deshalb wird der Pfad der Schlüsseldatei für Berechtigungsnachweise für

- Agent und Protokollfunktion, wird in der `output0.log`-Datei für diesen Agenten oder Logger protokolliert.
- Befehle in der Konsole angezeigt.

Die Java Systemeigenschaft **com.ibm.wqmfte.cred.keyfile** überschreibt alle anderen Eigenschaften. Ist die Systemeigenschaft nicht festgelegt, durchsucht der Agent die Datei mit dem ursprünglichen Schlüssel in der Datei `agent.properties` und anschließend in der Datei `installation.properties`.

Wenn die Datei mit dem ursprünglichen Schlüssel noch immer nicht gefunden wird und Sie den Schutzmodus im Befehl **fteObfuscate** auf 1 gesetzt haben, protokolliert der Agent in der Datei `output0.log` einen Fehler.

Wenn Sie den Schutzmodus im Befehl **fteObfuscate** auf `0` gesetzt haben, wird eine Warnung protokolliert, in der auf die Nichtweiterverwendung hingewiesen wird.

Die Protokollfunktion und der Befehl führen die gleichen Schritte beim Suchen der Datei mit dem ursprünglichen Schlüssel aus.

Protokollbridge und Connect:Direct-Bridge

Die Protokollbridge verwendet eine Eigenschaftendatei, `ProtocolBridgeProperties.xml`, zum Herstellen einer Verbindung zu FTP-, SFTP- und FTPS-Servern. Die Eigenschaftendatei enthält Verbindungsattribute, die erforderlich sind, um eine Verbindung zu diesen Servern herzustellen.

Wenn Sie den Wert der Attribute **credentialsFile** oder **credentialsKeyFile** in der Datei `ProtocolBridgeProperties.xml` ändern, muss der Bridgeagent neu gestartet werden.

Eines der Attribute lautet **credentialsFile** und der Wert enthält den Pfad zu einer XML-Datei mit der Benutzer-ID, dem Kennwort oder dem Schlüssel, damit eine Verbindung zu diesen Servern hergestellt werden kann. Der Standardwert für das Attribut ist *ProtocolBridgeCredentials.xml* und die Datei befindet sich in Ihrem Ausgangsverzeichnis, genau wie die `MQMFTCredentials.xml`-Datei.

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

Wie auch *MQMFTCredentails.xml* können Sie die Datei *ProtocolBridgeCredentials.xml* mit dem Befehl **fteObfuscate** verschlüsseln. Zur Entschlüsselung können Sie den erforderlichen Pfad zu einer Schlüsseldatei mit den Berechtigungsnachweisen mithilfe des zusätzlichen Elements **credentialsKeyFile** angeben, wie im folgenden Text gezeigt wird. Der Pfad kann Umgebungsvariablen enthalten.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Anmerkung: Die Angabe eines Werts für die Agenteneigenschaft **agentCredentialsKeyFile**, die Eigenschaft **commonCredentialsKeyFile** in `installation.properties` oder über die Systemeigenschaft **com.ibm.wqmfte.cred.keyfile** hat keine Auswirkung auf den Wert, der für das Attribut **credentialsKeyFile** angegeben ist.

Entsprechend verwendet die Connect:Direct-Bridge die Datei *ConnectDirectNodeProperties.xml*, um eine Verbindung zum Connect:Direct-Server herzustellen. Die XML-Datei enthält die erforderlichen Verbindungsinformationen sowie ein Attribut, das den Pfad zur XML-Datei mit den Berechtigungsnachweisen definiert. Die XML-Datei mit Berechtigungsnachweisen enthält eine Benutzer-ID oder ein Kennwort sowie zusätzliche Informationen, die erforderlich sind, um eine Verbindung zum Connect:Direct-Server herzustellen.

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

Wie auch die Datei *ProtocolBridgeCredentials.xml* können Sie *ConnectDirectCredentials.xml* mit dem Befehl **fteObfuscate** verschlüsseln. Zur Entschlüsselung können Sie den erforderlichen Pfad zu einer Schlüsseldatei mit den Berechtigungsnachweisen mithilfe des zusätzlichen Elements **credentialsKeyFile** angeben, wie im folgenden Text gezeigt wird. Der Pfad kann Umgebungsvariablen enthalten.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key”/>
```

Anmerkung: Die Angabe eines Werts für die Agenteneigenschaft **agentCredentialsKeyFile**, die Eigenschaft **commonCredentialsKeyFile** in `installation.properties` oder über die Systemeigenschaft **com.ibm.wqmfte.cred.keyfile** hat keine Auswirkung auf den Wert, der für das Attribut **credentialsKeyFile** angegeben ist.

Sie können das Element **credentialsKeyFile** angeben, ohne das Element **credentialsFile** in der Datei *ProtocolBridgeProperties.xml* anzugeben.

Wenn Sie das Element **credentialsFile** nicht angeben, wird die Standardberechtigungsnachweisdatei *ProtocolBridgeCredentials.xml* vom Protokollbridgeagenten verwendet und der Wert der Schlüsseldatei, der im Attribut **credentialsKeyFile** angegeben ist, wird zum Entschlüsseln der Berechtigungsnachweisdatei verwendet.

Entsprechend können Sie das Element **credentialsKeyFile** angeben, ohne das Element **credentialsFile** in der Datei *ConnectDirectNodeProperties.xml* anzugeben.

Wenn Sie das Element **credentialsFile** nicht angeben, wird die Standardberechtigungsnachweisdatei *ConnectDirectCredentials.xml* von der Connect:Direct-Bridge verwendet und der Wert der Schlüsseldatei, der im Attribut **credentialsKeyFile** angegeben ist, wird zum Entschlüsseln der Berechtigungsnachweisdatei verwendet.

Schlüssel aus dem Dataset unter z/OS verwenden

z/OS

Unter z/OS können Sie die Datei **MQMFTCredentials** angeben und die Schlüsseldatei mit den Berechtigungsnachweisen mithilfe eines PDSE (erweitertes partitioniertes Dataset) bereitstellen. Siehe „MQMFTCredentials.xml unter z/OS konfigurieren” auf Seite 616.

Zugehörige Verweise

Verbindung zwischen MFT-Befehlen und Warteschlangenmanagern

MFT-Berechtigungsnachweisdateiformat

fteObfuscate (sensible Daten verschlüsseln)

Verbindungsauthentifizierung für MFT und IBM MQ

Bei der Verbindungsauthentifizierung kann ein Warteschlangenmanager für die Authentifizierung von Anwendungen unter Verwendung einer angegebenen Benutzer-ID und eines angegebenen Kennworts konfiguriert werden. Wenn beim zugehörigen Warteschlangenmanager die Sicherheit aktiviert ist und er Berechtigungsnachweisdetails (Benutzer-ID und Kennwort) benötigt, muss die Verbindungsauthentifizierungsfunktion aktiviert sein, bevor eine erfolgreiche Verbindung zu einem Warteschlangenmanager hergestellt werden kann. Die Verbindungsauthentifizierung kann im Kompatibilitätsmodus oder im MQCSP-Authentifizierungsmodus ausgeführt werden.

Methoden zum Bereitstellen von Berechtigungsnachweisdetails

Viele Managed File Transfer-Befehle unterstützen die folgenden Methoden zum Bereitstellen von Berechtigungsnachweisdetails:

Details, die von Befehlszeilenargumenten bereitgestellt werden.
Die Berechtigungsnachweisdetails können mit den Parametern **-mquserid** und **-mqpassword** angegeben werden. Wenn **-mqpassword** nicht bereitgestellt wird, wird der Benutzer nach dem Kennwort gefragt; dabei wird die Eingabe nicht angezeigt.

Details, die aus einer Berechtigungsnachweisdatei bereitgestellt werden: MQMFTCredentials.xml.
Die Berechtigungsnachweisdetails können in einer `MQMFTCredentials.xml`-Datei entweder als Klartext oder als getrübter Text vordefiniert werden.

Informationen zur Einrichtung einer Datei MQMFTCredentials.xml unter IBM MQ for Multiplatforms finden Sie unter „MQMFTCredentials.xml auf Multiplatforms konfigurieren” auf Seite 614.

Informationen zur Einrichtung einer Datei MQMFTCredentials.xml unter IBM MQ for z/OS finden Sie unter „MQMFTCredentials.xml unter z/OS konfigurieren” auf Seite 616.

Vorrangstellung

Die Vorrangstellung bei der Bestimmung der Berechtigungsnachweisdetails lautet wie folgt:

1. Befehlszeilenargument.
2. `MQMFTCredentials.xml`-Index durch den zugeordneten Queue Manager und den Benutzer, der den Befehl ausführt.
3. `MQMFTCredentials.xml`-Index nach dem zugeordneten Queue Manager.
4. Standardmodus für Abwärtskompatibilität, bei dem keine Berechtigungsnachweisdetails angegeben werden, um Kompatibilität mit früheren Releases von IBM MQoder IBM WebSphere MQ zu ermöglichen

Anmerkungen:

- Die Befehle **fteStartAgent** und **fteStartLogger** unterstützen nicht das Befehlszeilenargument **-mquserid**oder **-mqpassword**und die Berechtigungsnachweisdetails können nur mit der Datei `MQMFTCredentials.xml` angegeben werden.
- z/OS

Unter z/OS muss das Kennwort in Großbuchstaben angegeben werden, selbst wenn das Benutzerkennwort Kleinbuchstaben enthält. Wenn das Kennwort des Benutzers z. B. "password" ist, muss es als "PASSWORD" eingegeben werden.

Zugehörige Verweise

Verbindung zwischen MFT-Befehlen und Warteschlangenmanagern

MFT-Berechtigungsnachweisdateiformat

`MQMFTCredentials.xml` auf Multiplatforms konfigurieren

Wenn Managed File Transfer (MFT) mit aktivierter Sicherheit konfiguriert ist, erfordert die Verbindungsauthentifizierung alle MFT -Befehle, die eine Verbindung mit einem Warteschlangenmanager herstellen, um die Benutzer-ID und das Kennwort anzugeben. Ebenso können MFT -Protokollfunktionen erforderlich sein, um eine Benutzer-ID und ein Kennwort anzugeben, wenn eine Verbindung zu einer Datenbank hergestellt wird. Diese Berechtigungsinformationen können in der Berechtigungsnachweisdatei MFT gespeichert werden.

Informationen zu diesem Vorgang

Die Elemente in der Datei `MQMFTCredentials.xml` müssen mit dem `MQMFTCredentials.xsd` -Schema übereinstimmen. Informationen zum Format von `MQMFTCredentials.xml`finden Sie unter Format der MFT-Berechtigungsnachweisdatei.

Eine Beispielberechtigungsnachweisdatei finden Sie im Verzeichnis `MQ_INSTALLATION_PATH/mqft/samples/credentials` .

Sie können über eine MFT-Berechtigungsnachweisdatei für den Koordinationswarteschlangenmanager, eine für den Befehlswarteschlangenmanager, eine für jeden Agenten und eine für jede Protokollfunktion verfügen. Alternativ können Sie eine Datei haben, die von allem in Ihrer Topologie verwendet wird.

Die Standardposition der MFT -Berechtigungsnachweisdatei lautet wie folgt:

Linux AIX **AIX and Linux**

$HOME

Windows **Windows**

%USERPROFILE% oder %HOMEDRIVE%%HOMEPATH%

Wenn die Berechtigungsnachweisdatei an einer anderen Position gespeichert ist, können Sie mit den folgenden Eigenschaften angeben, wo die Befehle nach ihr suchen sollen:

Tabelle 95. : Eigenschaften, die die Position der Datei MQMFTCredentials.xml für verschiedene Befehle definieren

Befehlstyp	Eigenschaftendatei	Eigenschaftsname
Befehl, der eine Verbindung zum Koordinationswarteschlangenmanager herstellt	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
Befehl, der eine Verbindung zum Befehlswarteschlangenmanager herstellt	connection.properties	connectionQMgrAuthenticationCredentialsFile
Befehl, der eine Verbindung zu einem Agentenprozess herstellt	agent.properties	agentQMgrAuthenticationCredentialsFile

Tabelle 95. : Eigenschaften, die die Position der Datei MQMFTCredentials.xml für verschiedene Befehle definieren (Forts.)		
Befehlstyp	**Eigenschaftendatei**	**Eigenschaftsname**
Befehl, der eine Verbindung zu einem Protokollfunktionsprozess herstellt	logger.properties	loggerQMgrAuthenticationCredentialsFile

Tabelle 96. : Eigenschaften, die die Position der Datei MQMFTCredentials.xml für Agenten und Protokollfunktionsprozesse definieren		
Befehlstyp	**Eigenschaftendatei**	**Eigenschaftsname**
MFT-Agenten	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT Protokollfunktionen	logger.properties	loggerQMgrAuthenticationCredentialsFile

Ausführliche Informationen dazu, welche Befehle und Prozesse eine Verbindung zu welchem Warteschlangenmanager herstellen, finden Sie unter Welche MFT -Befehle und -Prozesse eine Verbindung zu welchem Warteschlangenmanager herstellen.

V 9.2.0 V 9.2.0 Anstatt Eigenschaften in einzelnen Eigenschaftendateien hinzuzufügen, können Sie die Eigenschaft **commonCredentialsKeyFile** zur vorhandenen allgemeinen Datei `installation.properties` hinzufügen, damit Agent, Protokollfunktion und Befehle dieselbe Eigenschaft verwenden können.

Da die Berechtigungsnachweisdatei Benutzer-ID- und Kennwortinformationen enthält, sind spezielle Berechtigungen erforderlich, um unbefugten Zugriff darauf zu verhindern:

Linux AIX **AIX and Linux**

```
chown <agent owner userid>
chmod 600
```

Windows **Windows**

Stellen Sie sicher, dass die Übernahme nicht aktiviert ist, und entfernen Sie anschließend alle Benutzer-IDs, mit Ausnahme derjenigen, die den Agenten oder die Protokollfunktion ausführen, der bzw. die die Berechtigungsnachweisdatei verwendet.

Die Berechtigungsnachweisdetails, die für die Verbindung zu einem MFT -Koordinationswarteschlangenmanager im IBM MQ Explorer Managed File Transfer -Plug-in für verwendet werden, hängen vom Typ der Konfiguration ab:

Global (Konfiguration auf lokaler Platte)

Eine globale Konfiguration verwendet die Berechtigungsnachweisdatei, die in den Koordinations-und Befehlseigenschaften angegeben ist.

Lokal (definiert in IBM MQ Explorer):

Bei einer lokalen Konfiguration werden die Eigenschaften der Verbindungsdetails des zugehörigen Warteschlangenmanagers in IBM MQ Explorer verwendet.

Zugehörige Tasks

„Verbindungsauthentifizierung für MFT aktivieren” auf Seite 618

Eine Verbindungsauthentifizierung für das IBM MQ Explorer MFT-Plug-in, das mit einem Koordinationswarteschlangenmanager oder einem Befehlswarteschlangenmanager verbunden ist, und eine Verbindungsauthentifizierung für einen Managed File Transfer-Agenten, der mit einem Koordinationswarteschlangenmanager oder einem Befehlswarteschlangenmanager verbunden ist, kann im Kompatibilitätsmodus oder im MQCSP-Authentifizierungsmodus ausgeführt werden.

Zugehörige Verweise
MFT-Berechtigungsnachweisdateiformat
fteObfuscate: Verschlüsselung sensibler Daten

z/OS MQMFTCredentials.xml unter z/OS konfigurieren

Wenn Managed File Transfer (MFT) mit aktivierter Sicherheit konfiguriert ist, erfordert die Verbindungsauthentifizierung, dass alle MFT -Agenten und Befehle, die eine Verbindung zu einem Warteschlangenmanager herstellen, Berechtigungsnachweise für Benutzer-ID und Kennwort bereitstellen.

Ebenso können MFT -Protokollfunktionen erforderlich sein, um eine Benutzer-ID und ein Kennwort anzugeben, wenn eine Verbindung zu einer Datenbank hergestellt wird.

Diese Berechtigungsinformationen können in der Berechtigungsnachweisdatei MFT gespeichert werden. Beachten Sie, dass die Berechtigungsnachweisdateien optional sind. Es ist jedoch einfacher, die Dateien zu definieren, die Sie benötigen, bevor Sie die Umgebung anpassen.

Darüber hinaus erhalten Sie weniger Warnungen, wenn Sie über Berechtigungsnachweisdateien verfügen. Die Warnhinweise informieren Sie darüber, dass die Sicherheit des Warteschlangenmanagers laut MFT inaktiviert ist und Sie daher keine Authentifizierungsdaten angeben können.

Eine Beispielberechtigungsnachweisdatei finden Sie im Verzeichnis `MQ_INSTALLATION_PATH/mqft/samples/credentials`.

Es folgt ein Beispiel für eine Datei `MQMFTCredentials.xml`:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MFTCredentials.xsd">
  <tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
  <tns:qmgr name="MQPI" user="ADMIN" mqUserId="JOHNDOEI" mqPassword="yXXXX" />
  <tns:qmgr name="MQPH" mqUserId="NONEH" mqPassword="yXXXX" />
  <tns:qmgr name="MQPI" mqUserId="NONEI" mqPassword="yXXXX" />
</tns:mqmftCredentials>
```

Möchte ein Job mit der Benutzer-ID ADMIN eine Verbindung zum Warteschlangenmanager MQPH herstellen, übergibt er die Benutzer-ID *JOHNDOEH* und verwendet das Kennwort *cXXXX*.

Wenn der Job durch eine andere Benutzer-ID ausgeführt wird und eine Verbindung zu MQPH herstellt, übergibt dieser Job die Benutzer-ID *NONEH* und das Kennwort *yXXXX*.

Die Standardposition für die Datei `MQMFTCredentials.xml` ist das Ausgangsverzeichnis des Benutzers unter z/OS UNIX System Services (USS). Es ist auch möglich, die Datei in USS an einer anderen Position oder in einem Member innerhalb einer partitionierten Datei zu speichern.

Wenn die Berechtigungsnachweisdatei an einer anderen Position gespeichert ist, können Sie mit den folgenden Eigenschaften angeben, wo die Befehle nach ihr suchen sollen:

Tabelle 97. : Eigenschaften, die die Position der Datei MQMFTCredentials.xml für verschiedene Befehle definieren

Befehlstyp	Eigenschaftendatei	Eigenschaftsname
Befehl, der eine Verbindung zum Koordinationswarteschlangenmanager herstellt	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
Befehl, der eine Verbindung zum Befehlswarteschlangenmanager herstellt	connection.properties	connectionQMgrAuthenticationCredentialsFile

Tabelle 97. : Eigenschaften, die die Position der Datei MQMFTCredentials.xml für verschiedene Befehle definieren (Forts.)		
Befehlstyp	**Eigenschaftendatei**	**Eigenschaftsname**
Befehl, der eine Verbindung zu einem Agentenprozess herstellt	agent.properties	agentQMgrAuthenticationCredentialsFile
Befehl, der eine Verbindung zu einem Protokollfunktionsprozess herstellt	logger.properties	loggerQMgrAuthenticationCredentialsFile

Tabelle 98. : Eigenschaften, die die Position der Datei MQMFTCredentials.xml für Agenten und Protokollfunktionsprozesse definieren		
Befehlstyp	**Eigenschaftendatei**	**Eigenschaftsname**
MFT-Agenten	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT Protokollfunktionen	logger.properties	loggerQMgrAuthenticationCredentialsFile

Ausführliche Informationen dazu, welche Befehle und Prozesse eine Verbindung zu welchem Warteschlangenmanager herstellen, finden Sie unter Welche MFT -Befehle und -Prozesse eine Verbindung zu welchem Warteschlangenmanager herstellen.

Führen Sie die folgenden Schritte aus, um die Berechtigungsnachweisdatei in einer partitionierten Datei zu erstellen:

- Erstellen Sie eine PDSE mit dem Format VB und der logischen Satzlänge (Lrecl) 200.
- Erstellen Sie eine Teildatei im Dataset, notieren Sie sich Dataset und Teildatei und fügen Sie den folgenden Code zur Teildatei hinzu:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MQMFTCredentials.xsd">
  <!—-credentials information goes here-->
</tns:mqmftCredentials>
```

Sie können die Berechtigungsnachweisdatei mit einem Sicherheitsprodukt wie RACFschützen, aber die Benutzer-IDs, die die Managed File Transfer -Befehle ausführen und die Agenten-und Protokollfunktionsprozesse verwalten, benötigen Lesezugriff auf diese Datei.

Informationen in dieser Datei können mithilfe der JCL in der Teildatei BFGCROBS unkenntlich gemacht werden. Hierbei werden IBM MQ-Benutzer-ID und -Kennwort verschlüsselt. Unter Verwendung der Teildatei BFGCROBS wird aus der Zeile

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

beispielsweise die folgende erstellt:

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

Wenn Sie die Benutzer-ID für die Benutzer-ID-Zuordnung von IBM MQ behalten möchten, können Sie Kommentare zur Datei hinzufügen. Beispiel:

```
<!--      name="MQPI" user="ADMIN"    mqUserId="JOHNDOE1 -->
```

Diese Kommentare werden beim Unkenntlichmachen der Informationen nicht verändert.

Hinweis: Die Inhalte werden unkenntlich gemacht, nicht stark verschlüsselt. Daher ist es ratsam, die Anzahl der Benutzer-IDs mit Zugriff auf die Datei zu begrenzen.

Zugehörige Tasks

„MQMFTCredentials.xml auf Multiplatforms konfigurieren" auf Seite 614
Wenn Managed File Transfer (MFT) mit aktivierter Sicherheit konfiguriert ist, erfordert die Verbindungsauthentifizierung alle MFT -Befehle, die eine Verbindung mit einem Warteschlangenmanager herstellen, um die Benutzer-ID und das Kennwort anzugeben. Ebenso können MFT -Protokollfunktionen erforderlich sein, um eine Benutzer-ID und ein Kennwort anzugeben, wenn eine Verbindung zu einer Datenbank hergestellt wird. Diese Berechtigungsinformationen können in der Berechtigungsnachweisdatei MFT gespeichert werden.

Verbindungsauthentifizierung für MFT aktivieren

Eine Verbindungsauthentifizierung für das IBM MQ Explorer MFT-Plug-in, das mit einem Koordinationswarteschlangenmanager oder einem Befehlswarteschlangenmanager verbunden ist, und eine Verbindungsauthentifizierung für einen Managed File Transfer-Agenten, der mit einem Koordinationswarteschlangenmanager oder einem Befehlswarteschlangenmanager verbunden ist, kann im Kompatibilitätsmodus oder im MQCSP-Authentifizierungsmodus ausgeführt werden.

Informationen zu diesem Vorgang

Vor IBM MQ 9.2.0 ist der Kompatibilitätsmodus die Standardeinstellung für die Verbindungsauthentifizierung. Sie können den standardmäßig verwendeten Kompatibilitätsmodus aber auch inaktivieren und stattdessen den MQCSP-Authentifizierungsmodus aktivieren.

V 9.2.0 Ab IBM MQ 9.2.0 ist der MQCSP-Authentifizierungsmodus die Standardeinstellung.

Bei einer Verbindungsauthentifizierung für das IBM MQ Explorer Managed File Transfer-Plug-in oder für Managed File Transfer-Agenten, die eine Verbindung zu einem Warteschlangenmanager über den CLIENT-Transport herstellten, werden Kennwörter mit einer Länge von mehr als 12 Zeichen nur für den MQCSP-Authentifizierungsmodus unterstützt. Wenn Sie beim Authentifizieren mithilfe des Kompatibilitätsmodus ein Kennwort mit einer Länge von mehr als 12 Zeichen angeben, tritt ein Fehler auf und der Agent kann den Warteschlangenmanager nicht authentifizieren. Weitere Informationen finden Sie in der BFGAG0187E-Nachricht unter Diagnosenachrichten: BFGAG0001 - BFGAG9999.

Prozedur

- Um einen Verbindungsauthentifizierungsmodus für einen Koordinationswarteschlangenmanager oder einen Befehlswarteschlangenmanager in IBM MQ Explorer auszuwählen, gehen Sie folgendermaßen vor:
  a) Wählen Sie den Warteschlangenmanager aus, zu dem eine Verbindung hergestellt werden soll.
  b) Klicken Sie mit der rechten Maustaste, und wählen Sie im Kontextmenü **Verbindungsdetails -> Eigenschaften** aus.
  c) Klicken Sie auf die Registerkarte **Benutzer-ID** .
  d) Stellen Sie sicher, dass das Kontrollkästchen für den Verbindungsauthentifizierungsmodus ausgewählt ist, den Sie verwenden möchten:
     – V 9.1.0 Ab IBM MQ 9.1.0 ist das Kontrollkästchen **Benutzer-ID-Kompatibilitätsmodus** standardmäßig abgewählt. Wenn also das Kontrollkästchen **Benutzer-ID aktivieren** ausgewählt ist, verwendet der IBM MQ Explorer beim Herstellen einer Verbindung zum Warteschlangenmanager die MQCSP-Authentifizierung. Wenn IBM MQ Explorer eine Verbindung zum Warteschlangenmanager nicht mit der MQCSP-Authentifizierung, sondern mithilfe des Kompatibilitätsmodus herstellen muss, stellen Sie sicher, dass sowohl das Kontrollkästchen **Benutzer-ID aktivieren** als auch das Kontrollkästchen **Benutzer-ID-Kompatibilitätsmodus** ausgewählt ist.
     – Vor IBM MQ 9.1.0 ist dass Kontrollkästchen **Benutzer-ID-Kompatibilitätsmodus** standardmäßig ausgewählt. Wenn also das Kontrollkästchen **Benutzer-ID aktivieren** ausgewählt ist, verwendet

der IBM MQ Explorer beim Herstellen einer Verbindung zum Warteschlangenmanager den Kompatibilitätsmodus. Wenn IBM MQ Explorer eine Verbindung zum Warteschlangenmanager mithilfe der MQCSP-Authentifizierung herstellen muss, stellen Sie sicher dass das Kontrollkästchen **Benutzer-ID aktivieren** ausgewählt und das Kontrollkästchen **Benutzer-ID-Kompatibilitätsmodus** abgewählt ist.

- Um den MQCSP-Authentifizierungsmodus für einen Managed File Transfer-Agenten mithilfe der Datei `MQMFTCredentials.xml` zu aktivieren oder zu inaktivieren, fügen Sie den Parameter **useMQCSPAuthentication** der Datei `MQMFTCredentials.xml` für den betreffenden Benutzer hinzu.

  Der Parameter **useMQCSPAuthentication** hat die folgenden Werte:

  true
  Der MQCSP-Authentifizierungsmodus wird zur Authentifizierung des Benutzers beim Warteschlangenmanager verwendet.

  V 9.2.0 Ab IBM MQ 9.2.0 ist `true` der Standardwert. Wenn der Parameter **useMQCSPAuthentication** nicht angegeben ist, wird er standardmäßig auf `true` gesetzt und der MQCSP-Authentifizierungsmodus wird zur Authentifizierung des Benutzers beim Warteschlangenmanager verwendet.

  false
  Der Kompatibilitätsmodus wird zur Authentifizierung des Benutzers beim Warteschlangenmanager verwendet.

  Wenn vor IBM MQ 9.2.0 der Parameter **useMQCSPAuthentication** nicht angegeben ist, wird er standardmäßig auf `false` gesetzt und der Kompatibilitätsmodus wird zur Authentifizierung des Benutzers beim Warteschlangenmanager verwendet.

  Das folgende Beispiel zeigt, wie der Parameter **useMQCSPAuthentication** in der Datei `MQMFTCredentials.xml` festgelegt wird:

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

Zugehörige Konzepte

„MQCSP-Kennwortschutz” auf Seite 34
Ab IBM MQ 8.0 können Sie Kennwörter, die in der MQCSP-Struktur enthalten sind, mit der IBM MQ-Funktion geschützt oder durch die TLS-Verschlüsselung verschlüsselt senden.

Zugehörige Verweise

„Verbindungsauthentifizierung für MFT und IBM MQ” auf Seite 613
Bei der Verbindungsauthentifizierung kann ein Warteschlangenmanager für die Authentifizierung von Anwendungen unter Verwendung einer angegebenen Benutzer-ID und eines angegebenen Kennworts konfiguriert werden. Wenn beim zugehörigen Warteschlangenmanager die Sicherheit aktiviert ist und er Berechtigungsnachweisdetails (Benutzer-ID und Kennwort) benötigt, muss die Verbindungsauthentifizierungsfunktion aktiviert sein, bevor eine erfolgreiche Verbindung zu einem Warteschlangenmanager hergestellt werden kann. Die Verbindungsauthentifizierung kann im Kompatibilitätsmodus oder im MQCSP-Authentifizierungsmodus ausgeführt werden.

MFT-Berechtigungsnachweisdateiformat

MFT-Sandboxes

Sie können den Bereich des Dateisystems einschränken, auf den der Agent als Teil einer Übertragung zugreifen kann. Der Bereich, der für den Agenten eingeschränkt ist, wird als Sandbox bezeichnet. Sie können Einschränkungen auf den Agenten anwenden oder auf den Benutzer, der eine Übertragung anfordert.

Wenn es sich bei dem Agenten um einen Protokollbridgeagenten oder einen Connect:Direct-Bridgeagenten handelt, werden keine Sandboxes unterstützt. Die Sandbox-Funktion kann nicht für Agenten verwendet werden, die Übertragungen zu oder von IBM MQ-Warteschlangen ausführen.

Zugehörige Verweise

„Mit Sandboxes für MFT-Agenten arbeiten” auf Seite 620

Sie können Managed File Transfer noch weiter absichern, indem Sie den Bereich eines Dateisystems einschränken, auf den ein Agent zugreifen kann.

„Mit MFT-Benutzersandboxes arbeiten” auf Seite 621
Sie können den Bereich des Dateisystems einschränken, in das Dateien auf der Basis des MQMD-Benutzernamens, der die Übertragung anfordert, in das und aus dem Dateisystem übertragen werden können.

Mit Sandboxes für MFT-Agenten arbeiten

Sie können Managed File Transfer noch weiter absichern, indem Sie den Bereich eines Dateisystems einschränken, auf den ein Agent zugreifen kann.

Die Sandbox-Funktion kann nicht für Agenten verwendet werden, die Übertragungen an oder von IBM MQ-Warteschlangen durchführen. Die Einschränkung des Zugriffs auf IBM MQ-Warteschlangen durch Sandboxing kann stattdessen durch die Benutzersandbox-Funktion, die empfohlene Lösung für alle Sandboxing-Anforderungen, implementiert werden. Weitere Informationen zur Benutzersandbox-Funktion finden Sie im Abschnitt „Mit MFT-Benutzersandboxes arbeiten” auf Seite 621

Zum Aktivieren des Sandboxing des Agenten fügen Sie die folgende Eigenschaft zur Datei `agent.properties` für den Agenten hinzu, den Sie beschränken möchten:

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

Dabei gilt:

- *restricted_directory_name* ist ein Verzeichnispfad, der zugelassen oder verweigert werden soll.
- `!` ist optional und gibt an, dass der folgende Wert für *restricted_directory_name* verweigert wird (ausgeschlossen). Wenn `!` nicht angegeben ist, ist *restricted_directory_name* ein zulässiger (eingeschlossene) Pfad.
- *separator* ist das plattformspezifische Trennzeichen.

Wenn Sie beispielsweise den Zugriff, den AGENT1 nur für das Verzeichnis `/tmp` hat, beschränken möchten, aber nicht zulassen, dass auf das Unterverzeichnis `private` zugegriffen werden kann, setzen Sie die Eigenschaft wie folgt in der `agent.properties`-Datei, die zu AGENT1: `sandboxRoot=/tmp:!/tmp/private`gehört.

Die Eigenschaft 'sandboxRoot' wird im Abschnitt Erweiterte Agenteneigenschaften beschrieben.

Weder die Agentensandbox- noch die Benutzersandbox-Funktion werden auf Protokollbridgeagenten oder Connect:Direct-Bridgeagenten unterstützt.

Auf AIX, Linux, and Windows-Plattformen in einer Sandbox arbeiten

ALW Auf AIX, Linux, and Windows-Plattformen schränkt Sandboxing die Verzeichnisse ein, die ein Managed File Transfer Agent lesen und schreiben kann. Wird die Sandbox-Funktion aktiviert, hat der Managed File Transfer Agent Lese- und Schreibzugriff auf die angegebenen Verzeichnisse sowie auf alle Unterverzeichnisse, sofern ihm in 'sandboxRoot' der Zugriff nicht verweigert wurde. Das Sandbox-Verfahren von Managed File Transfer hat keinen Vorrang vor der Sicherheitseinstellung des Betriebssystems. Der Benutzer, von dem der Managed File Transfer Agent gestartet wurde, muss über einen geeigneten Verzeichniszugriff auf Betriebssystemebene verfügen, um Lese- und Schreibvorgänge für das Verzeichnis ausführen zu können. Einer symbolischen Verbindung zu einem Verzeichnis wird nicht gefolgt, wenn sich dieses Verzeichnis außerhalb der angegebenen sandboxRoot-Verzeichnisse (und -Unterverzeichnisse) befindet.

Unter z/OS in einer Sandbox arbeiten

z/OS Unter z/OS werden mit dem Sandbox-Verfahren die Qualifikationsmerkmale des Dataset-Namens beschränkt, die dem Managed File Transfer Agent für Lese- und Schreibvorgänge zur Verfügung stehen. Der Benutzer, von dem der Managed File Transfer Agent gestartet wurde, muss über geeignete

Betriebssystemberechtigungen für die involvierten Datasets verfügen. Wenn Sie einen sandboxRoot-Wert für das Qualifikationsmerkmal des Dataset-Namens in Anführungszeichen setzen, entspricht der Wert der üblichen z/OS-Konvention und wird als vollständig qualifizierter Wert behandelt. Wenn Sie die Anführungszeichen weglassen, wird sandboxRoot die aktuelle Benutzer-ID als Präfix vorangestellt. Wenn Sie beispielsweise die Eigenschaft sandboxRoot auf `sandboxRoot=//test`setzen, kann der Agent auf die folgenden Datasets zugreifen (in der Standardnotation z/OS). `//`*`username`*`.test.**` Wenn die Anfangsebenen des vollständig aufgelösten Datasetnamens nicht mit dem sandboxRootübereinstimmen, wird die Übertragungsanforderung zurückgewiesen.

Auf IBM i-Systemen in einer Sandbox arbeiten

IBM i Für das integrierte Dateisystem von IBM i-Systemen werden mit der Sandbox-Funktion die Verzeichnisse eingeschränkt, auf die ein Managed File Transfer Agent Lese- bzw. Schreibzugriff hat. Wird die Sandbox-Funktion aktiviert, hat der Managed File Transfer Agent Lese- und Schreibzugriff auf die angegebenen Verzeichnisse sowie auf alle Unterverzeichnisse, sofern ihm in 'sandboxRoot' der Zugriff nicht verweigert wurde. Das Sandbox-Verfahren von Managed File Transfer hat keinen Vorrang vor der Sicherheitseinstellung des Betriebssystems. Der Benutzer, von dem der Managed File Transfer Agent gestartet wurde, muss über einen geeigneten Verzeichniszugriff auf Betriebssystemebene verfügen, um Lese- und Schreibvorgänge für das Verzeichnis ausführen zu können. Einer symbolischen Verbindung zu einem Verzeichnis wird nicht gefolgt, wenn sich dieses Verzeichnis außerhalb der angegebenen sandbox-Root-Verzeichnisse (und -Unterverzeichnisse) befindet.

Zugehörige Verweise

„Zusätzliche Prüfungen für Platzhalterübertragungen” auf Seite 624
Wenn ein Agent mit einer Benutzer-oder Agentensandbox konfiguriert wurde, um die Positionen zu beschränken, an die der Agent Dateien übertragen kann, und von, können Sie angeben, dass für diesen Agenten zusätzliche Prüfungen auf Platzhalterzeichen durchgeführt werden sollen.

„Mit Sandboxes für MFT-Agenten arbeiten” auf Seite 620
Sie können Managed File Transfer noch weiter absichern, indem Sie den Bereich eines Dateisystems einschränken, auf den ein Agent zugreifen kann.

Die MFT `agent.properties`-Datei

Mit MFT-Benutzersandboxes arbeiten

Sie können den Bereich des Dateisystems einschränken, in das Dateien auf der Basis des MQMD-Benutzernamens, der die Übertragung anfordert, in das und aus dem Dateisystem übertragen werden können.

Benutzersandboxes werden nicht unterstützt, wenn es sich bei dem Agenten um einen Protokollbridgeagenten oder Connect:Direct-Bridgeagenten handelt.

Zum Aktivieren des Benutzer-Sandboxings fügen Sie die folgende Eigenschaft zur Datei `agent.properties` für den Agenten hinzu, den Sie beschränken möchten:

```
userSandboxes=true
```

Wenn diese Eigenschaft vorhanden und auf "wahr" gesetzt ist, verwendet der Agent die Informationen in der Datei *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name*`/agents/`*agent_name*`/UserSandboxes.xml`, um festzustellen, auf welche Teile des Dateisystems der Benutzer zugreifen kann, der die Übertragung anfordert.

Die Datei `UserSandboxes.xml` setzt sich aus einem <agent>-Element zusammen, das null oder mehr <sandbox>-Elemente enthält. Diese Elemente beschreiben, welche Regeln auf welche Benutzer angewendet werden. Das Attribut `user` des Elements <sandbox> ist ein Muster, das zum Abgleich mit dem MQMD-Benutzer der Anforderung verwendet wird.

Die Datei `UserSandboxes.xml` wird vom Agenten regelmäßig erneut geladen, und alle gültigen Änderungen an der Datei wirken sich auf das Verhalten des Agenten aus. Standardmäßig erfolgt die Neuladung alle 30 Sekunden. Dieses Intervall kann durch Angabe der Agenteneigenschaft 'xmlConfigReloadInterval' in der Datei '`agent.properties`' geändert werden.

Bei Angabe des Attributs oder Werts `userPattern="regex"` wird das Attribut `user` als regulärer Java-Ausdruck interpretiert. Weitere Informationen finden Sie im Abschnitt Von MFT verwendete reguläre Ausdrücke.

Wenn Sie das Attribut `userPattern="regex"` nicht angeben, wird das Attribut `user` als Muster mit folgenden Platzhalterzeichen interpretiert:

- Stern (*), der null oder mehr Zeichen darstellt
- Fragezeichen (?), das genau ein Zeichen darstellt

Die Übereinstimmungen werden in der Reihenfolge ausgeführt, in der die `<sandbox>`-Elemente in der Datei aufgelistet sind. Nur die erste Übereinstimmung wird verwendet, alle folgenden potenziellen Übereinstimmungen in der Datei werden ignoriert. Wenn keines der in der Datei angegebenen `<sandbox>`-Elemente mit dem MQMD-Benutzer übereinstimmt, der der Übertragungsanforderungsnachricht zugeordnet ist, kann die Übertragung nicht auf das Dateisystem zugreifen. Wenn eine Übereinstimmung zwischen dem MQMD-Benutzernamen und einem Attribut `user` gefunden wurde, gibt die Übereinstimmung eine Gruppe von Regeln in einem Element `<sandbox>` an, die auf die Übertragung angewendet werden. Diese Gruppe von Regeln wird verwendet, um festzustellen, von welchen Dateien oder Dateigruppen als Teil der Übertragung gelesen oder in diese geschrieben werden kann.

Jede Gruppe von Regeln kann ein `<read>`-Element angeben, das angibt, welche Dateien gelesen werden können, sowie ein `<write>`-Element, das angibt, welche Dateien geschrieben werden können. Wenn Sie die `<read>` oder `<write>`-Elemente aus einer Gruppe von Regeln weglassen, wird davon ausgegangen, dass der Benutzer, der dieser Gruppe von Regeln zugeordnet ist, keine Lese- oder Schreibvorgänge durchführen darf.

Anmerkung: Das `<read>`-Element muss vor dem `<write>`-Element stehen, und das `<include>`-Element muss sich vor dem `<exclude>`-Element in der Datei `UserSandboxes.xml` befinden.

Jedes `<read>` oder `<write>`-Element enthält eines oder mehrere Muster, die verwendet werden, um zu bestimmen, ob sich eine Datei in der Sandbox befindet und übertragen werden kann. Geben Sie diese Muster an, indem Sie die Elemente `<include>` und `<exclude>` verwenden. Das `name`-Attribut des `<include>`- oder `<exclude>`-Elements gibt das Muster an, das abgeglichen werden soll. Ein optionales Attribut `type` gibt an, ob der Namenswert ein Datei-oder Warteschlangenmuster ist. Wenn das Attribut `type` nicht angegeben wird, behandelt der Agent das Muster als Datei-oder Verzeichnispfadmuster. Beispiel:

```
<tns:read>
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

Die Muster `<include>` und `<exclude>` `name` werden vom Agenten verwendet, um zu bestimmen, ob Dateien, Datasets oder Warteschlangen gelesen oder geschrieben werden können. Eine Operation ist zulässig, wenn der kanonische Dateipfad , der Datensatz oder der Warteschlangenname mit mindestens einem der eingeschlossenen Muster und genau null der ausgeschlossenen Muster übereinstimmt. Die Muster, die mit dem Attribut `name` der Elemente `<include>` und `<exclude>` angegeben werden, verwenden die Pfadtrennzeichen und Konventionen, die für die Plattform, auf der der Agent ausgeführt wird, geeignet sind. Wenn Sie relative Dateipfade angeben, werden die Pfade in Bezug auf die `transferRoot` -Eigenschaft des Agenten aufgelöst.

Wenn Sie eine Warteschlangeneinschränkung angeben, wird die Syntax QUEUE@QUEUEMANAGER mit den folgenden Regeln unterstützt:

- Wenn das Zeichen (@) im Eintrag fehlt, wird das Muster wie ein Warteschlangenname behandelt, auf den auf jedem WS-Manager zugegriffen werden kann. Wenn es sich bei dem Muster beispielsweise um `name` handelt, wird die gleiche Weise wie `name@**` behandelt.
- Wenn das Zeichen (@) das erste Zeichen im Eintrag ist, wird das Muster als Warteschlangenmanagername und alle Warteschlangen auf dem WS-Manager behandelt. Wenn es sich bei dem Muster beispielsweise um `@name` handelt, wird die gleiche Weise wie `**@name` behandelt.

Die folgenden Platzhalterzeichen haben eine besondere Bedeutung, wenn Sie sie im Attribut name der Elemente <include> und <exclude> angeben:

*****
Ein einzelner Stern entspricht null oder mehr Zeichen in einem Verzeichnisnamen oder in einem Qualifikationsmerkmal eines Dataset-oder Warteschlangennamens.

?
Ein Fragezeichen entspricht genau einem Zeichen in einem Verzeichnisnamen oder in einem Qualifikationsmerkmal eines Dataset-oder Warteschlangennamens.

******
Zwei Sterne entsprechen null oder mehr Verzeichnisnamen oder null oder mehr Qualifikationsmerkmalen in einem Dateinamen oder Warteschlangennamen. Darüber hinaus haben Pfade, die mit einem Pfadtrennzeichen enden, ein implizites " ** " am Ende des Pfads hinzugefügt. /home/user/ ist also dasselbe wie /home/user/**.

Beispiel:

- /**/test/** stimmt mit jeder Datei überein, die über ein test-Verzeichnis in ihrem Pfad verfügt.
- /test/file? stimmt mit jeder Datei innerhalb des /test-Verzeichnisses überein, die mit der Zeichenfolge file beginnt, gefolgt von einem einzelnen Zeichen.
- c:\test*.txt stimmt mit jeder Datei im c:\test-Verzeichnis mit einer .txt-Erweiterung überein
- c:\test***.txt stimmt mit der Datei innerhalb des Verzeichnisses 'c:\test oder eines seiner Unterverzeichnisse überein, die eine Erweiterung .txt aufweist.
- z/OS //'TEST.*.DATA' stimmt mit jedem Datensatz überein, der das erste Qualifikationsmerkmal von TEST, ein zweites Qualifikationsmerkmal und ein drittes Qualifikationsmerkmal von DATA hat.
- *@QM1 stimmt mit jeder Warteschlange auf dem WS-Manager QM1 überein, die ein einzelnes Qualifikationsmerkmal hat.
- TEST.*.QUEUE@QM1 stimmt mit einer beliebigen Warteschlange auf dem Queue Manager QM1 überein, der das erste Qualifikationsmerkmal von TEST, ein zweites Qualifikationsmerkmal und ein drittes Qualifikationsmerkmal von QUEUE hat.
- **@QM1 stimmt mit allen Warteschlangen auf dem Warteschlangenmanager QM1überein.

Symbolische Links

Sie müssen alle symbolischen Links, die Sie in Dateipfaden in der UserSandboxes.xml-Datei verwenden, vollständig auflösen, indem Sie feste Verbindungen in den Elementen <include> und <exclude> angeben. Wenn Sie beispielsweise einen symbolischen Link haben, bei dem /var /SYSTEM/varzugeordnet wird, müssen Sie diesen Pfad als <tns:include name="/SYSTEM/var"/>angeben. Andernfalls schlägt die beabsichtigte Übertragung mit einem Benutzer-Sandbox-Sicherheitsfehler fehl.

Beispiel

Dieses Beispiel zeigt, wie der Benutzer mit dem MQMD-Benutzernamen guest jede Datei aus dem Verzeichnis /home/user/public oder einem seiner Unterverzeichnisse auf dem System, auf dem der Agent AGENT_JUPITER ausgeführt wird, übertragen kann, indem das folgende Element <sandbox> zur Datei UserSandboxes.xml im Konfigurationsverzeichnis von AGENT_JUPITER hinzugefügt wird:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
        xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="guest">
            <tns:read>
                <tns:include name="/home/user/public/**"/>
```

```
            </tns:read>
        </tns:sandbox>
    </tns:agent>
</tns:userSandboxes>
```

Beispiel

In diesem Beispiel wird gezeigt, wie einem beliebigen Benutzer mit dem MQMD-Benutzernamen `account` gefolgt von einer einzigen Ziffer, z. B. `account4`, die folgenden Aktionen ausgeführt werden können:

- Übertragen Sie eine beliebige Datei aus dem Verzeichnis `/home/account` oder einem der zugehörigen Unterverzeichnisse, außer dem Verzeichnis `/home/account/private` auf dem System, auf dem der Agent AGENT_SATURN ausgeführt wird.
- Übertragen Sie eine beliebige Datei in das `/home/account/output`-Verzeichnis oder in ein beliebiges seiner Unterverzeichnisse auf dem System, auf dem der Agent AGENT_SATURN ausgeführt wird.
- Lesen Sie die Nachrichten aus Warteschlangen auf dem lokalen Queue Manager, die mit dem Präfix `ACCOUNT.` beginnen, es sei denn, sie beginnt mit `ACCOUNT.PRIVATE.` (das heißt, `PRIVATE` auf der zweiten Ebene).
- Übertragen Sie Daten in Warteschlangen, die mit dem Präfix `ACCOUNT.OUTPUT.` beginnen, auf einem beliebigen Queue Manager.

Damit ein Benutzer mit dem MQMD-Benutzernamen `account` diese Aktionen ausführen kann, fügen Sie das folgende Element `<sandbox>` zur Datei `UserSandboxes.xml`im Konfigurationsverzeichnis von AGENT_SATURN hinzu:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
            xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
            xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="account[0-9]" userPattern="regex">
            <tns:read>
                <tns:include name="/home/account/**"/>
                <tns:include name="ACCOUNT.**" type="queue"/>
                <tns:exclude name="ACCOUNT.PRIVATE.**" type="queue"/>
                <tns:exclude name="/home/account/private/**"/>
                                                    </tns:read>
            <tns:write>
                <tns:include name="/home/account/output/**"/>
                <tns:include name="ACCOUNT.OUTPUT.**" type="queue"/>
            </tns:write>
        </tns:sandbox>
    </tns:agent>
</tns:userSandboxes>
```

Zugehörige Verweise

„Zusätzliche Prüfungen für Platzhalterübertragungen” auf Seite 624
Wenn ein Agent mit einer Benutzer-oder Agentensandbox konfiguriert wurde, um die Positionen zu beschränken, an die der Agent Dateien übertragen kann, und von, können Sie angeben, dass für diesen Agenten zusätzliche Prüfungen auf Platzhalterzeichen durchgeführt werden sollen.

Die MFT `agent.properties`-Datei

Zusätzliche Prüfungen für Platzhalterübertragungen

Wenn ein Agent mit einer Benutzer-oder Agentensandbox konfiguriert wurde, um die Positionen zu beschränken, an die der Agent Dateien übertragen kann, und von, können Sie angeben, dass für diesen Agenten zusätzliche Prüfungen auf Platzhalterzeichen durchgeführt werden sollen.

additionalWildWildcardSandboxChecking (Eigenschaft)

Um eine zusätzliche Prüfung für Platzhalterübertragungen zu aktivieren, fügen Sie die folgende Eigenschaft der `agent.properties`-Datei für den Agenten hinzu, den Sie überprüfen möchten.

```
additionalWildcardSandboxChecking=true
```

Wenn diese Eigenschaft auf "true" gesetzt ist und der Agent eine Übertragungsanforderung vornimmt, die versucht, eine Position zu lesen, die sich außerhalb der definierten Sandbox für die Dateiübereinpassung des Platzhalterzeichens befindet, schlägt die Übertragung fehl. Wenn eine Übertragungsanforderung aus mehreren Übertragungen besteht und eine dieser Übertragungen fehlschlägt, weil sie versucht, eine Position außerhalb der Sandbox zu lesen, schlägt die gesamte Übertragung fehl. Wenn die Prüfung fehlschlägt, wird die Fehlerursache in einer Fehlernachricht angezeigt.

Wenn die Eigenschaft 'additionalWildcardSandboxChecking' aus der `agent.properties`-Datei eines Agenten weggelassen wird oder auf 'falsch' gesetzt ist, werden keine zusätzlichen Prüfungen für Platzhalterübertragungen für diesen Agenten durchgeführt.

Fehlernachrichten für die Überprüfung auf Platzhalterzeichen

Die Nachrichten, die gemeldet werden, wenn eine Anforderung zum Übertragen von Platzhalterzeichen an einen Standort außerhalb einer konfigurierten Sandbox-Position gestellt wird, lauten wie folgt.

Die folgende Nachricht tritt auf, wenn sich ein Platzhalterdateipfad in einer Übertragungsanforderung außerhalb der eingeschränkten Sandbox befindet:

```
BFGSS0077E: Es wurde versucht, den Dateipfad zu lesen:  path  wurde verweigert.
Der Dateipfad befindet sich außerhalb der Sandbox mit eingeschränkter Übertragung.
```

Die folgende Nachricht tritt auf, wenn eine Übertragung innerhalb einer Anforderung mit mehreren Übertragungsanweisungen eine Anforderung mit Platzhalterzeichen enthält, bei der sich der Pfad außerhalb der eingeschränkten Sandbox befindet:

```
BFGSS0078E: Es wurde versucht, den Dateipfad zu lesen:  path  wurde als andere Übertragung ignoriert.
-Element in der verwalteten Übertragung versuchte, außerhalb der Sandbox mit eingeschränkter Übertragung zu lesen.
```

Die folgende Nachricht tritt auf, wenn sich eine Datei außerhalb der eingeschränkten Sandbox befindet:

```
BFGSS0079E: Der Versuch, die Datei  file path  zu lesen, wurde verweigert.
Die Datei befindet sich außerhalb der Sandbox mit eingeschränkter Übertragung.
```

Die folgende Nachricht wird in einer Mehrfach-Übertragungsanforderung ausgegeben, bei der eine andere Anforderung mit Platzhalterzeichen die folgende Nachricht verursacht hat:

```
BFGSS0080E: Es wurde versucht, die Datei zu lesen:  file path  wurde als andere Übertragung ignoriert.
-Element in der verwalteten Übertragung versuchte, außerhalb der Sandbox mit eingeschränkter Übertragung zu lesen.
```

Bei Einzeldateiübertragungen, die keine Platzhalterzeichen enthalten, wird die Nachricht, die bei der Übertragung gemeldet wird, eine Datei, die sich außerhalb der Sandbox befindet, nicht von früheren Releases geändert:

```
Fails with BFGIO0056E: Der Versuch, die Datei "FILE" zu lesen, wurde verweigert.
Die Datei befindet sich außerhalb der Sandbox mit eingeschränkter Übertragung.
```

Zugehörige Verweise

„Mit MFT-Benutzersandboxes arbeiten” auf Seite 621
Sie können den Bereich des Dateisystems einschränken, in das Dateien auf der Basis des MQMD-Benutzernamens, der die Übertragung anfordert, in das und aus dem Dateisystem übertragen werden können.

„Mit Sandboxes für MFT-Agenten arbeiten” auf Seite 620
Sie können Managed File Transfer noch weiter absichern, indem Sie den Bereich eines Dateisystems einschränken, auf den ein Agent zugreifen kann.

Die MFT `agent.properties`-Datei

SSL- oder TLS-Verschlüsselung für MFT konfigurieren

Sie können SSL oder TLS mit IBM MQ Managed File Transfer verwenden, um die Kommunikation zwischen Agenten und ihren Agentenwarteschlangenmanagern, den Befehlen und den Warteschlangenmanagern, zu denen sie eine Verbindung herstellen, sowie den verschiedenen Verbindungen zwischen Warteschlangenmanagern in Ihrer Topologie zu sichern.

Vorbereitende Schritte

Sie können die SSL-oder TLS-Verschlüsselung verwenden, um Nachrichten zu verschlüsseln, die durch eine IBM MQ Managed File Transfer -Topologie fließen. Hierzu gehören folgende Aufrufe:

- Nachrichten, die zwischen einem Agenten und seinem Agentenwarteschlangenmanager übergeben werden.
- Nachrichten für Befehle und die Warteschlangenmanager, zu denen sie eine Verbindung herstellen.
- Interne Nachrichten, die zwischen den Agenten-WS-Managern, Befehlswarteschlangenmanagern und Koordinationswarteschlangenmanagern innerhalb der Topologie fließen.

Informationen zu diesem Vorgang

Allgemeine Informationen zur Verwendung von SSL mit IBM MQ finden Sie unter „Mit SSL/TLS arbeiten” auf Seite 294. In IBM MQ ist Managed File Transfer die als Standard verwendete Java-Clientanwendung.

Führen Sie die folgenden Schritte aus, um SSL mit Managed File Transfer zu verwenden:

Vorgehensweise

1. Erstellen Sie eine Truststore-Datei und optional eine Schlüsselspeicherdatei (diese Dateien können die gleiche Datei sein). Wenn Sie keine Clientauthentifizierung (d. h. SSLCAUTH=OPTIONAL auf Kanälen) benötigen, müssen Sie keinen Keystore bereitstellen. Sie benötigen nur einen Truststore, um das Zertifikat des Warteschlangenmanagers zu authentifizieren.

   Der Schlüsselalgorithmus, der für die Erstellung von Zertifikaten für den Truststore und die Keystores verwendet wird, muss RSA sein, damit Sie mit IBM MQarbeiten können.
2. Richten Sie Ihren IBM MQ-Warteschlangenmanager für die Verwendung von SSL ein.

   Informationen zur Konfiguration eines Warteschlangenmanagers für die Verwendung von SSL (z. B. mit IBM MQ Explorer) finden Sie im Abschnitt SSL für Warteschlangenmanager konfigurieren.
3. Speichern Sie die Truststore-Datei und die Schlüsselspeicherdatei (falls vorhanden) an einer geeigneten Position. Eine empfohlene Position ist das Verzeichnis *config_directory*/*coordination_qmgr*/agents/*agent_name*.
4. Legen Sie die SSL-Eigenschaften wie erforderlich für jeden SSL-fähigen Warteschlangenmanager in der entsprechenden Managed File Transfer-Eigenschaftendatei fest. Jede Gruppe von Eigenschaften bezieht sich auf einen separaten Warteschlangenmanager (Agent, Koordination und Befehl), obwohl ein WS-Manager zwei oder mehr dieser Rollen ausführen kann.

   Eine der Eigenschaften **CipherSpec** oder **CipherSuite** ist erforderlich, andernfalls versucht der Client, eine Verbindung ohne SSL herzustellen. Aufgrund der Terminologieunterschiede zwischen IBM MQ und Javawerden die Eigenschaften **CipherSpec** und **CipherSuite** bereitgestellt. Da Managed File Transfer beide Eigenschaften akzeptiert und die erforderliche Konvertierung vornimmt, müssen Sie nicht beide Eigenschaften setzen. Wenn Sie sowohl die **CipherSpec** -als auch die **CipherSuite** -Eigenschaften angeben, hat **CipherSpec** Vorrang.

   Die Eigenschaft **PeerName** ist optional. Sie können die Eigenschaft auf den definierten Namen des Warteschlangenmanagers setzen, zu dem Sie eine Verbindung herstellen wollen. Managed File Transfer lehnt Verbindungen mit einem falschen SSL-Server mit einem unpassenden definierten Namen ab.

   Legen Sie die Eigenschaften für **SslTrustStore** und **SslKeyStore** auf Dateinamen fest, die auf die Truststore-und Schlüsselspeicherdateien verweisen. Wenn Sie diese Eigenschaften für einen Agenten einrichten, der bereits aktiv ist, stoppen Sie den Agenten, und starten Sie ihn erneut, um die Verbindung im SSL-Modus wieder herzustellen.

Eigenschaftendateien enthalten Plain-Text-Kennwörter. Daher sollten Sie die Festlegung geeigneter Dateisystemberechtigungen in Betracht ziehen.

Weitere Informationen zu SSL-Eigenschaften finden Sie im Abschnitt „SSL/TLS-Eigenschaften für MFT" auf Seite 627.

5. Wenn ein Agentenwarteschlangenmanager SSL verwendet, können Sie die erforderlichen Details beim Erstellen des Agenten nicht angeben. Führen Sie die folgenden Schritte aus, um den Agenten zu erstellen:
   a) Erstellen Sie den Agenten mit dem Befehl **fteCreateAgent** . Sie erhalten eine Warnung, dass es nicht möglich ist, das Vorhandensein des Agenten im Koordinations-WS-Manager zu veröffentlichen.
   b) Bearbeiten Sie die `agent.properties`-Datei, die im vorherigen Schritt erstellt wurde, um die SSL-Informationen hinzuzufügen. Wenn der Agent erfolgreich gestartet wurde, wird die Publizierung erneut versucht.
6. Wenn Agenten oder Instanzen des IBM MQ-Explorers ausgeführt werden, während die SSL-Eigenschaften in der `agent.properties`-Datei oder in der `coordination.properties`-Datei geändert werden, müssen Sie den Agenten oder IBM MQ Explorer erneut starten.

Zugehörige Verweise
Die MFT `agent.properties`-Datei

SSL/TLS-Eigenschaften für MFT

Einige MFT-Eigenschaftendateien enthalten SSL- und TLS-Eigenschaften. Mit SSL oder TLS können Sie in IBM MQ und Managed File Transfer unberechtigte Verbindungen zwischen Agenten und Warteschlangenmanagern verhindern und die Nachrichtenübertragungen zwischen Agenten und Warteschlangenmanagern verschlüsseln.

Folgende MFT-Eigenschaftendateien enthalten SSL-Eigenschaften:

- SSL/TLS-Eigenschaften für die MFT `agent.properties`-Datei
- SSL/TLS-Eigenschaften für die MFT `coordination.properties`-Datei
- SSL/TLS-Eigenschaften für die MFT `command.properties`-Datei
- SSL/TLS-Eigenschaften für die MFT `logger.properties`-Datei

Informationen zur Verwendung von SSL oder TLS mit Managed File Transfer finden Sie unter SSL- oder TLS-Verschlüsselung für MFT konfigurieren.

Ab IBM WebSphere MQ 7.5 können Sie Umgebungsvariablen in einigen Managed File Transfer-Eigenschaften verwenden, die Datei- oder Verzeichnispositionen darstellen. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt). Weitere Informationen finden Sie unter Die Verwendung von Umgebungsvariablen in MFT-Eigenschaften.

Verbindung zu einem WS-Manager im Clientmodus mit Kanalauthentifizierung herstellen

In IBM WebSphere MQ 7.1 wurden Kanalauthentifizierungsdatensätze eingeführt, um den Zugriff auf Kanalebene steuern zu können. Diese Änderung bringt es mit sich, dass neu erstellte Warteschlangenmanager in IBM WebSphere MQ 7.1 oder höheren Versionen Clientverbindungen der Komponente Managed File Transfer ablehnen.

Weitere Informationen zur Kanalauthentifizierung finden Sie im Abschnitt „Kanalauthentifizierungssätze" auf Seite 54.

Ist in der Kanalauthentifizierungskonfiguration für die von Managed File Transfer verwendete Serververbindung (SVRCONN) eine nicht privilegierte MCAUSER-ID angegeben, müssen Sie für die Warteschlangenmanager, Warteschlangen und Themen spezifische Berechtigungssätze angeben, damit der Managed File Transfer Agent und Befehle fehlerfrei funktionieren. Verwenden Sie den MQSC-Befehl SET CHLAUTH oder

den PCF-Befehl Set Channel Authentication Record , um Kanalauthentifizierungsdatensätze zu erstellen, zu ändern oder zu entfernen. Für Managed File Transfer-Agenten, die eine Verbindung zu IBM WebSphere MQ 7.1-Warteschlangenmanagern (oder höher) herstellen sollen, kann eine gemeinsame MCAUSER-ID oder jeweils eine eigene MCAUSER-ID eingerichtet werden.

Erteilen Sie jeder MCAUSER-ID die folgenden Berechtigungen:

- Berechtigungssätze, die für den Warteschlangenmanager erforderlich sind:
  - Verbinden
  - setid
  - inq
- Berechtigungsdatensätze, die für Warteschlangen erforderlich sind.

  Für alle agentenspezifischen Warteschlangen (d. h. mit Warteschlangennamen in der folgenden Liste, an deren Ende *Agentenname* angehängt ist) müssen diese Warteschlangen-Berechtigungssätze für jeden Agenten erstellt werden, der zum Warteschlangenmanager in IBM WebSphere MQ 7.1 oder höher eine Clientverbindung herstellen soll.
  - put, get, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)
  - put, get, setid, durchsuchen (SYSTEM.FTE.COMMAND. *agent_name*)
  - put, get (SYSTEM.FTE.DATA. *agent_name*)
  - put, get (SYSTEM.FTE.REPLY. *agent_name*)
  - put, get, inq, durchsuchen (SYSTEM.FTE.STATE. *agent_name*)
  - put, get, browse (SYSTEM.FTE.EVENT. *agent_name*)
  - put, get (SYSTEM.FTE)
- Berechtigungssätze, die für Themen erforderlich sind:
  - sub, pub (SYSTEM.FTE)
- Für Dateiübertragungen erforderliche Berechtigungsdatensätze.

  Wenn Sie über separate MCAUSER-IDs für Quellen-und Zielagent verfügen, erstellen Sie die Berechtigungsdatensätze in den Warteschlangen der Agenten an der Quelle und an der Zieladresse.

  Beispiel: Wenn die MCAUSER-ID des Quellenagenten **user1** und die MCAUSER-ID des Zielagenten **user2** ist, legen Sie die folgenden Berechtigungen für die Agentenbenutzer fest:

Agentbenutzer	Warteschlange	Berechtigung erforderlich
user1	SYSTEM.FTE.DATA *destination_agent_name*	put
user1	SYSTEM.FTE.COMMAND. *destination_agent_name*	put
user2	SYSTEM.FTE.REPLY. *source_agent_name*	put
user2	SYSTEM.FTE.COMMAND. *source_agent_name*	put

SSL oder TLS zwischen dem Connect:Direct-Bridgeagenten und dem Connect:Direct-Knoten konfigurieren

Sie können den Connect:Direct-Bridgeagenten und den Connect:Direct-Knoten so konfigurieren, dass die Verbindung zwischen beiden über das SSL-Protokoll hergestellt wird. Dazu müssen Sie einen Keystore und einen Truststore erstellen und Einstellungen in der Eigenschaftendatei des Connect:Direct-Bridgeagenten vornehmen.

Informationen zu diesem Vorgang

Diese Schritte enthalten Anweisungen zum Abrufen der Schlüssel, die von einer Zertifizierungsstelle signiert wurden. Wenn Sie keine Zertifizierungsstelle verwenden, können Sie ein selbst signiertes Zertifikat

generieren. Weitere Informationen über das Generieren eines selbst signierten Zertifikats finden Sie unter „Mit SSL/TLS unter AIX, Linux, and Windows arbeiten” auf Seite 307.

Die nachfolgenden Schritte enthalten Anweisungen zur Erstellung eines neuen Keystore und Truststore für den Connect:Direct-Bridgeagenten. Wenn der Connect:Direct-Bridgeagent bereits einen Keystore und Truststore für die sichere Verbindung mit IBM MQ-Warteschlangenmanagern verwendet, können Sie den vorhandenen Keystore und Truststore auch für die sichere Verbindung mit dem Connect:Direct-Knoten verwenden. Weitere Informationen finden Sie unter „SSL- oder TLS-Verschlüsselung für MFT konfigurieren” auf Seite 626.

Vorgehensweise

Führen Sie für den Connect:Direct-Knoten die folgenden Schritte aus:

1. Generieren Sie einen Schlüssel und ein signiertes Zertifikat für den Connect:Direct-Knoten.

   Hierfür können Sie das Tool IBM Key Management verwenden, das mit IBM MQbereitgestellt wird. Weitere Informationen finden Sie unter „Mit SSL/TLS arbeiten” auf Seite 294.
2. Senden Sie eine Anforderung an eine Zertifizierungsstelle, um den Schlüssel signiert zu haben. Sie erhalten ein Zertifikat im Gegenzug.
3. Erstellen Sie eine Textdatei (z. B. `/test/ssl/certs/CAcert`), die den öffentlichen Schlüssel Ihrer Zertifizierungsinstanz enthält.
4. Installieren Sie die Option Secure+ auf dem Connect:Direct-Knoten.

   Wenn der Knoten bereits vorhanden ist, können Sie die Secure + Option installieren, indem Sie das Installationsprogramm erneut ausführen. Geben Sie dabei die Position der vorhandenen Installation an, und wählen Sie nur die Option Secure + (Sicherheit) aus.
5. Erstellen Sie eine neue Textdatei (z. B. `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`).
6. Kopieren Sie das Zertifikat, das Sie von Ihrer Zertifizierungsstelle erhalten haben, und den privaten Schlüssel, der sich in `/test/ssl/cd/privateKeys/`*`node_name`*`.key` befindet, in die Textdatei.

   Der Inhalt von `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt` muss das folgende Format haben:

```
-----BEGIN CERTIFICATE-----
MIICnzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJHQjES
MBAGA1UECBMJSGFtcHNoaXJlMRAwDgYDVQQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJ
Qk0xDjAMBgNVBAsTBU1RSVBUMQswCQYDVQQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAxCzAJBgNVBAYTAkdCMRIwEAYDVQQIEwlIYW1wc2hp
cmUxDDAKBgNVBAoTA0lCTTEOMAwGA1UECxMFTVFGVEUxDzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAvgP1QIklU9ypSKD1XoODo1yk
EyMFXBOUpZRrDVxjoSEC0vtWNcJ199e+Vc4UpNybDyBu+NkDlMNofX4QxeQcLAFj
WnhakqCiQ+JIAD5Aurhnrwche0MV3kjA84GKH/rOSVqtl984mu/lDyS819XcfSSn
cOOMsK1KbneVSCIV2XECAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhCAQ0E
HxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFNXMIpSc
csBXUniW4A3UrZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj4lVz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAFc7klXa4pGKYgwchxKpE3ZF6FNwy4vBXS216/ja
8h/vl8+iv01OCL8t0ZOKSU95fyZLzOPKnCH7v+ItFSE3CIiEk9Dlz2U6WO9lICwn
l7PL72TdfaL3kabwHYVf17IVcuL+VZsZ3HjLggP2qHO9ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxLOJ/gRUOIQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDWoMNt5fj51v7aPmVeS60bOm+UlGre8B/Ze18JVj2O4K2Uh72rDCXE
5e6eFxSdUM207sQDy20euBVELJtM2kOkL1ROdoQQSlU3XQNgJw/t3ZIx5hPXWEQT
rjRQO64BEhb+PzzxPF8uwzZ9IrUK9BJ/UUnqC6OdBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI8OGvBvWXO4fHyvIX5aslwhBoArXIS1AtNTrptPvoaP1zyIAeZ6OCVo/
SFo+A2UhmtEJeOJaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWDyJBVhUOjXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfDwP+bEjDzUaaarJTS7lIFeLLw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLKOfYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+lYcX8OVZ6
sHFPNlHluCNy/riUcBy9iviVeodX8IomOchSyO5DKl8bwZNjYtUP+CtYHNFU5BaD
I+1uUOAeJ+wjQYKT1WaeIGZ3VxuNITJul8y5qDTXXfX7vxM5OoWXa6U5+AYuGUMg
/itPZmUmNrHjTk7ghT6i1IQOaBowXXKJBlMmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50loLnBRqWbfR+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP7lzQ==
-----END RSA PRIVATE KEY-----
```

7. Starten Sie das Secure + Admin Tool.

- Führen Sie auf AIX and Linux-Systemen den Befehl **spadmin.sh** aus.
- Klicken Sie auf Windows-Systemen auf **Start** > **Programme** > **Sterling Commerce Connect:Direct** > **CD Secure+ Admin Tool**.

Das CD Secure + Admin Tool wird gestartet.

8. Klicken Sie im CD Secure + Admin Tool doppelt auf die Zeile **.Local** , um die Haupt-SSL-oder TLS-Einstellungen zu bearbeiten.
   a) Wählen Sie **Enable SSL Protocol** oder **Enable TLS Protocol** (TLS-Protokoll aktivieren) aus, je nachdem, welches Protokoll Sie verwenden.
   b) Wählen Sie **Überschreibungsüberschreibung inaktivieren** aus.
   c) Wählen Sie mindestens eine Cipher Suite aus.
   d) Wenn Sie eine bidirektionale Authentifizierung wünschen, ändern Sie den Wert für **Clientauthentifizierung aktivieren** in `Yes`.
   e) Geben Sie im Feld **Vertrauenswürdiges Stammzertifikat** den Pfad zur öffentlichen Zertifikatsdatei Ihrer Zertifizierungsstelle ein, `/test/ssl/certs/CAcert`.
   f) Geben Sie im Feld **Schlüsselzertifikatsdatei** den Pfad zu der Datei ein, die Sie erstellt haben, `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`.
9. Klicken Sie doppelt auf die Zeile **.Client** , um die Haupt-SSL-oder TLS-Einstellungen zu bearbeiten.
   a) Wählen Sie **Enable SSL Protocol** oder **Enable TLS Protocol** (TLS-Protokoll aktivieren) aus, je nachdem, welches Protokoll Sie verwenden.
   b) Wählen Sie **Überschreibungsüberschreibung inaktivieren** aus.

Führen Sie für den Connect:Direct-Bridgeagenten die folgenden Schritte aus:

10. Erstellen Sie einen Truststore. Sie können dies tun, indem Sie einen Dummy-Schlüssel erstellen und dann den Dummy-Schlüssel löschen.

    Sie können die folgenden Befehle verwenden:

    ```
    keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
    ```

    ```
    keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
    ```

11. Importieren Sie das öffentliche Zertifikat der Zertifizierungsstelle in den Truststore.

    Sie können den folgenden Befehl verwenden:

    ```
    keytool -import -trustcacerts -alias myCA
            -file /test/ssl/certs/CAcert
            -keystore /test/ssl/fte/stores/truststore.jks
    ```

12. Bearbeiten Sie die Eigenschaftendatei des Connect:Direct-Bridgeagenten.

    Fügen Sie die folgenden Zeilen an einer beliebigen Position in die Datei ein:

    ```
    cdNodeProtocol=protocol
    cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
    cdNodeTruststorePassword=password
    ```

    Im Beispiel in diesem Schritt ist *protocol* das Protokoll, das Sie verwenden, entweder SSL oder TLS, und *password* ist das Kennwort, das Sie bei der Erstellung des Truststores angegeben haben.

13. Wenn Sie eine beidseitige Authentifizierung wünschen, erstellen Sie einen Schlüssel und ein Zertifikat für den Connect:Direct-Bridgeagenten.
    a) Erstellen Sie einen Schlüsselspeicher und einen Schlüssel.

       Sie können den folgenden Befehl verwenden:

       ```
       keytool -genkey -keyalg RSA -alias agent_name
       ```

```
        -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -validity 365
```

b) Erstellen Sie eine Signieranforderung.

Sie können den folgenden Befehl verwenden:

```
keytool -certreq -v -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks -storepass password
        -file /test/ssl/fte/requests/agent_name.request
```

c) Importieren Sie das Zertifikat, das Sie von dem vorhergehenden Schritt erhalten haben, in den Keystore. Das Zertifikat muss im Format x.509 angegeben werden.

Sie können den folgenden Befehl verwenden:

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -file certificate_file_path
```

d) Bearbeiten Sie die Eigenschaftendatei des Connect:Direct-Bridgeagenten.

Fügen Sie die folgenden Zeilen an einer beliebigen Position in die Datei ein:

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

In dem Beispiel in diesem Schritt ist *password* das Kennwort, das Sie beim Erstellen des Keystores angegeben haben.

Zugehörige Tasks

Connect:Direct-Bridge konfigurieren

ALW AMQP-Clients schützen

Sie verwenden eine Reihe von Sicherheitsmechanismen, um Verbindungen von AMQP-Clients zu sicher und zu gewährleisten, das die Daten im Netz in geeigneter Weise geschützt sind. Sie können Sicherheit in Ihre MQ Light-Anwendungen integrieren. Sie können die vorhandenen Sicherheitsfunktionen von IBM MQ auch auf die gleiche Weise mit AMQP-Clients verwenden, mit der die Funktionen für andere Anwendungen verwendet werden.

Kanalauthentifizierungsregeln (CHLAUTH)

Mit den Kanalauthentifizierungsregeln können Sie TCP-Verbindungen auf einen Warteschlangenmanager beschränken. AMQP-Kanäle unterstützen die Verwendung von Kanalauthentifizierungsregeln, die Sie für Ihren Warteschlangenmanager konfigurieren. Wenn Kanalauthentifizierungsregeln mit einem Profil definiert werden, das mit AMQP-Kanälen in Ihrem Warteschlangenmanager übereinstimmt, werden die Regeln für diese Kanäle angewendet. Standardmäßig ist die Kanalauthentifizierung bei neuen IBM® MQ-Warteschlangenmanagern aktiviert, d. h., Sie müssen zumindest einige Konfigurationen durchführen, bevor Sie einen AMQP-Kanal verwenden können.

Weitere Informationen zur Konfiguration von Kanalauthentifizierungsregeln für die Aktivierung von AMQP-Verbindungen für Ihre Warteschlangenmanager finden Sie unter AMQP-Kanäle erstellen und verwenden.

Verbindungsauthentifizierung (CONNAUTH)

Mit der Verbindungsauthentifizierung können Sie Verbindungen zu einem Warteschlangenmanager authentifizieren. AMQP-Kanäle unterstützen die Verwendung der Verbindungsauthentifizierung, um den Zugriff auf die Warteschlangenmanager aus AMQP-Anwendungen zu steuern.

Das AMQP-Protokoll verwendet das SASL-Framework (Simple Authentication and Security Layer), um anzugeben, wie eine Verbindung authentifiziert wird. Es gibt verschiedene SASL-Verfahren und IBM MQ unterstützt zwei davon: ANONYMOUS und PLAIN.

Bei der Verwendung von ANONYMOUS werden keine Berechtigungsnachweise vom Client zur Authentifizierung an den Warteschlangenmanager übergeben. Wenn das im Attribut CONNAUTH angegebene MQ-Objekt AUTHINFO den CHCKCLNT-Wert REQUIRED oder REQDADM hat (bei der Verbindung als Benutzer mit Verwaltungsaufgaben), wird die Verbindung zurückgewiesen. Wenn CHCKCLNT den Wert NONE oder OPTIONAL hat, wird die Verbindung akzeptiert.

Bei der Verwendung von PLAIN wird ein Benutzername und ein Kennwort vom Client zur Authentifizierung an den Warteschlangenmanager übergeben. Wenn das im Attribut CONNAUTH angegebene MQ-Objekt AUTHINFO den CHCKCLNT-Wert NONE hat, wird die Verbindung zurückgewiesen. Wenn CHCKCLNT den Wert OPTIONAL, REQUIRED oder REQDADM hat (bei der Verbindung als Benutzer mit Verwaltungsaufgaben), wird der Benutzername und das Kennwort vom Warteschlangenmanager geprüft. Der Warteschlangenmanager überprüft das Betriebssystem (wenn das Objekt AUTHINFO den Typ IDPWOS hat) oder ein LDAP-Repository (wenn das Objekt AUTHINFO den Typ IDPWLDAP hat).

In der folgenden Tabelle wird diese Authentifizierung zusammengefasst:

Tabelle 99. Zusammenfassung von SASL-Verfahren und Verbindungsauthentifizierung

SASL-Verfahren	**Werden Berechtigungsnachweise vom Client an den Warteschlangenmanager übergeben?**	**CHKCLNT-Wert**
ANONYMOUS	Nein	REQUIRED oder REQDADM - Verbindung zurückgewiesen NONE oder OPTIONAL - Verbindung akzeptiert
PLAIN	Ja, Benutzername und Kennwort	REQUIRED, REQDADM oder OPTIONAL - Benutzername und Kennwort werden vom Warteschlangenmanager geprüft NONE - Verbindung zurückgewiesen

Wenn Sie einen MQ Light-Client verwenden, können Sie Berechtigungsnachweise angeben, indem Sie diese in die AMQP-Adresse integrieren, zu der Sie eine Verbindung herstellen; Beispiel:

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

MCAUSER-Einstellung in einem Kanal

AMQP-Kanäle verfügen über das Attribut MCAUSER, mit dem die IBM MQ-Benutzer-ID festgelegt werden kann, unter der alle Verbindungen zu diesem Kanal berechtigt sind. Alle Verbindungen von AMQP-Clients zu diesem Kanal übernehmen die MCAUSER-ID, die Sie konfiguriert haben. Mit dieser Benutzer-ID wird die Nachrichtenübermittlung zu verschiedenen Themen autorisiert.

Es wird empfohlen, Verbindungen zu Warteschlangenmanagern mit der Kanalauthentifizierung (CHLAUTH) zu sichern. Wenn Sie die Kanalauthentifizierung verwenden, wird empfohlen, den Wert von MCAUSER für einen nicht privilegierten Benutzer zu konfigurieren. Wenn eine Verbindung zu einem Kanal nicht mit einer CHLAUTH-Regel übereinstimmt, wird somit sichergestellt, dass mit dieser Verbindung keine Nachrichtenübertragung an den Warteschlangenmanager ausgeführt werden kann.

Anmerkung: Windows Unter Windows wird die Festlegung der MCAUSER-Benutzer-ID vor IBM MQ 9.1.1 nur für Benutzer-IDs mit einer Länge von bis zu 12 Zeichen unterstützt. V 9.2.0 Ab IBM MQ 9.1.1 Continuous Delivery und ab IBM MQ 9.2.0 Long Term Support ist die Begrenzung auf 12 Zeichen entfernt.

SSL/TLS-Unterstützung

AMQP-Kanäle unterstützen die SSL/TLS-Verschlüsselung mithilfe von Schlüsseln aus dem Schlüsselrepository, das für Ihren Warteschlangenmanager konfiguriert ist. Die Konfigurationsoptionen für AMQP-Kanäle zur SSL/TLS-Verschlüsselung unterstützen die gleichen Optionen wie andere Typen von MQ-Kanälen. Sie können eine Verschlüsselungsspezifikation angeben und festlegen, ob für den Warteschlangenmanager Zertifikate aus AMQP-Clientverbindungen erforderlich sind.

Durch die Verwendung der FIPS-Attribute des Warteschlangenmanagers können Sie die SSL/TLS-CipherSuites steuern, mit denen sichere Verbindungen von AMQP-Clients hergestellt werden können.

Weitere Informationen zum Einrichten eines Schlüsselrepositorys für den Warteschlangenmanager finden Sie unter Mit SSL oder TLS auf UNIX-, Linux- und Windows-Systemen arbeiten.

Weitere Informationen zum Konfigurieren der SSL/TLS-Unterstützung für eine AMQP-Clientverbindung finden Sie unter AMQP-Kanäle erstellen und verwenden.

Java Authentication and Authorization Service (JAAS)

Sie können optional AMQP-Kanäle mit einem JAAS-Anmeldemodul konfigurieren, mit dem der von einem AMQP-Client bereitgestellte Benutzername und das zugehörige Kennwort geprüft werden können. Weitere Informationen finden Sie unter „JAAS für AMQP-Kanäle konfigurieren” auf Seite 634.

Zugehörige Tasks

AMQP-Clientanwendungen entwickeln

AMQP-Kanäle erstellen und verwenden

ALW Übernahme von AMQP-Clients beschränken

Wenn eine AMQP-Clientverbindung hergestellt wird, in der die gleiche Client-ID wie in einer vorhandenen AMQP-Clientverbindung verwendet wird, wird die vorhandene Clientverbindung standardmäßig getrennt. Sie können den Warteschlangenmanager allerdings so konfigurieren, dass das Übernahmeverhalten des Client eingeschränkt wird, damit die Übernahme nur möglich ist, wenn bestimmte Kriterien erfüllt werden.

Wenn beispielsweise AMQP-Anwendungen von verschiedenen Teams entwickelt werden und dabei zufälligerweise die gleiche Kunden-ID verwendet wird, ist es möglicherweise nicht sinnvoll, die vorhandene Clientverbindung zu trennen. Um dieses Problem zu vermeiden, können Sie die Clientübernahme auf Basis des Namens des verwendeten AMQP-Kanals, der IP-Adresse des Clients und der Client-Benutzer-ID (wenn die SASL-Authentifizierung aktiviert ist) einschränken.

Verwenden Sie die Einstellungen der Warteschlangenmanagerattribute **AdoptNewMCA** und **AdoptNewMCACheck**, um die erforderliche Ebene für die Einschränkungen bei der Clientübernahme anzugeben, wie in der folgenden Tabelle beschrieben wird:

*Tabelle 100. Einstellungen **AdoptNewMCA** und **AdoptNewMCACheck** für die Einschränkung bei der Clientübernahme*

AdoptNewMCA	AdoptNewMCACheck	Kriterien, die vor der Clientübernahme überprüft werden
NO oder nicht definiert	Nicht zutreffend	Keine. Die Clientübernahme ist für alle Clientverbindungen zulässig, die authentifiziert sind und alle CHLAUTH-Regeln bestehen.
ALL (oder ein anderer Wert als NO)	QM oder nicht definiert	Keine. Die Clientübernahme ist für alle Clientverbindungen zulässig, die authentifiziert sind und alle CHLAUTH-Regeln bestehen.

*Tabelle 100. Einstellungen **AdoptNewMCA** und **AdoptNewMCACheck** für die Einschränkung bei der Clientübernahme (Forts.)*

AdoptNewMCA	AdoptNewMCACheck	Kriterien, die vor der Clientübernahme überprüft werden
ALL (oder ein anderer Wert als NO)	NAME	Benutzer-ID (wenn SASL aktiviert ist) Kanalname
ALL (oder ein anderer Wert als NO)	ADDRESS	Benutzer-ID (wenn SASL aktiviert ist) IP-Adresse
ALL (oder ein anderer Wert als NO)	ALLE	Benutzer-ID (wenn SASL aktiviert ist) Kanalname IP-Adresse

Die Warteschlangenmanagerattribute **AdoptNewMCA** und **AdoptNewMCACheck** sind Teil der Warteschlangenmanagerkonfiguration, wie in der Zeilengruppe CHANNELS definiert ist. Ändern Sie auf IBM MQ for Windows- und IBM MQ for Linux x86-64-Systemen die Konfigurationsinformationen mit dem IBM MQ Explorer. Ändern Sie auf anderen Systemen die Informationen, indem Sie die `qm.ini`-Konfigurationsdatei bearbeiten. Informationen zum Ändern der Kanalinformationen des Warteschlangenmanagers finden Sie unter Kanalattribute.

Zugehörige Tasks

AMQP-Clientanwendungen entwickeln

AMQP-Kanäle erstellen und verwenden

ALW JAAS für AMQP-Kanäle konfigurieren

Die angepassten JAAS-Module (Java Authentication and Authorization Service) können zur Authentifizierung der Berechtigungsnachweise für Benutzernamen und Kennwörter verwendet werden, die von einem AMQP-Client beim Herstellen einer Verbindung an einen AMQP-Kanal übergeben werden.

Informationen zu diesem Vorgang

Sie können ein angepasstes JAAS-Modul verwenden, wenn Sie bereits JAAS-Module für die Authentifizierung in anderen Java-basierten Systemen verwenden und diese Module für die Authentifizierung von AMQP-Verbindungen zu MQ wiederverwenden möchten. Alternativ können Sie ein angepasstes JAAS-Modul schreiben, wenn die in MQ integrierten Authentifizierungsfunktionen das Authentifizierungsverfahren, das Sie verwenden möchten, nicht unterstützen.

Die Konfiguration von JAAS-Modulen für AMQP-Kanäle wird auf Warteschlangenmanagerebene ausgeführt. Dadurch wird das Modul bei der Konfiguration eines JAAS-Moduls für die Authentifizierung von AMQP-Verbindungen mit dem Warteschlangenmanager für alle AMQP-Kanäle angewendet. Der Name des Kanals, der das JAAS-Modul aufgerufen hat, wird an das Modul übergeben, damit Sie ein unterschiedliches Anmeldeverhalten für verschiedene Kanäle codieren können.

Es werden auch weitere Informationen an das JAAS-Modul übergeben:

- Die Client-ID des AMQP-Clients, der versucht, eine Authentifizierung durchzuführen.
- Die Netzadresse des AMQP-Clients.
- Der Name des Kanals, der das JAAS-Modul aufgerufen hat.

Vorgehensweise

Mit den folgenden Schritten können Sie ein JAAS-Konfigurationsmodul für AMQP-Kanäle konfigurieren:

1. Definieren Sie eine `jaas.config`-Datei, die eine oder mehrere Zeilengruppen für die JAAS-Modulkonfiguration enthält. Die Zeilengruppe muss den vollständig qualifizierten Namen der Java-Klasse angeben, mit der die JAAS-Schnittstelle `javax.security.auth.spi.LoginModule` implementiert wird.
   - Eine `jaas.config`-Standarddatei wird mit dem Produkt geliefert und befindet sich in *QM_data_directory*`/amqp/jaas.config`.
   - In der Standarddatei `jaas.config` ist bereits eine vorkonfigurierte Zeilengruppe mit der Bezeichnung `MQXRConfig` definiert.
2. Geben Sie den Namen der Zeilengruppe an, die für AMQP-Kanäle verwendet werden soll.
   - Linux AIX Fügen Sie der `amqp_unix.properties`-Datei eine Eigenschaft hinzu.
   - Windows Fügen Sie der `amqp_win.properties`-Datei eine Eigenschaft hinzu.

   Die Eigenschaft hat das folgende Format:

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   Beispiel:

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```
3. Konfigurieren Sie die Umgebung des Warteschlangenmanagers, um die Klasse des angepassten Moduls einzuschließend. Der AMQP-Service muss Zugriff auf die in der JAAS-Konfigurationszeilengruppe konfigurierte Java-Klasse haben.

   Dazu fügen Sie den Pfad der JAAS-Klasse zur MQ `service.env`-Datei hinzu. Bearbeiten Sie die `service.env`-Datei im MQ-Konfigurationsverzeichnis (*MQ_config_directory*) oder im Konfigurationsverzeichnis des Queue Managers (*QM_config_directory*), um die Variable CLASSPATH auf die Position der JAAS-Modulklasse zu setzen.

Nächste Schritte

Ein Beispiel für ein JAAS-Anmeldemodul wird mit dem Produkt im Verzeichnis *mq_installation_directory*`/amqp/samples` geliefert. Mit dem Beispiel für ein JAAS-Anmeldemodul werden alle Clientverbindungen authentifiziert, unabhängig vom Benutzernamen oder Kennwort, mit denen der Client eine Verbindung herstellt.

Sie können den Quellcode des Beispiels ändern und ihn erneut kompilieren, damit die Authentifizierung nur für bestimmte Benutzer mit einem bestimmten Kennwort vorgenommen wird. Gehen Sie folgendermaßen vor, um den AMQP-Kanal auf einem UNIX-System für die Verwendung des Beispiels für ein JAAS-Anmeldemodul zu konfigurieren, das mit dem Produkt geliefert wird:

1. Bearbeiten Sie die Datei `/var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties` und legen Sie die Eigenschaft `com.ibm.mq.MQXR.JAASConfig=MQXRConfig` fest.
2. Bearbeiten Sie die Datei /var/mqm/service.env und legen Sie die Eigenschaft CLASSPATH=*mq_installation_location*`/amqp/samples` fest.

Die Datei `jaas.config` enthält bereits eine Zeilengruppe mit dem Namen `MQXRConfig` , die die Beispielklasse `samples.JAASLoginModule` als Anmeldemodulklasse angibt. Für das Testen des Beispielmoduls sind keine Änderungen an `jaas.config` erforderlich.

Zugehörige Tasks

AMQP-Clientanwendungen entwickeln

AMQP-Kanäle erstellen und verwenden

Advanced Message Security

Advanced Message Security (AMS) ist eine Komponente von IBM MQ, die ein hohes Maß an Schutz für sensible Daten bereitstellt, die über das IBM MQ-Netz fließen, während die Endanwendungen nicht beeinflusst werden.

Überblick über Advanced Message Security

IBM MQ-Anwendungen können Advanced Message Security verwenden, um sensible Daten (z. B. Finanztransaktionen mit hohem Wert und persönliche Informationen) mit unterschiedlichen Schutzstufen zu senden, indem Sie ein Verschlüsselungsmodell mit öffentlichen Schlüsseln verwenden.

Zugehörige Verweise

GSKit-Rückgabecode, die in AMS-Nachrichten verwendet werden

Funktionen von Advanced Message Security

Mit Advanced Message Security werden die IBM MQ-Sicherheitsservices um die Bereitstellung der Signatur und Verschlüsselung von Daten auf Nachrichtenebene erweitert. Die erweiterten Services stellen sicher, dass die Nachrichtendaten nicht geändert wurden, wenn sie ursprünglich in eine Warteschlange gestellt wurden und wenn sie abgerufen werden. Außerdem stellt AMS sicher, dass ein Sender von Nachrichtendaten berechtigt ist, signierte Nachrichten in eine Zielwarteschlange zu stellen.

AMS stellt die folgenden Funktionen bereit:

- Sichert sensible oder hochwertige Transaktionen, die von IBM MQ verarbeitet werden.
- Erkennt und entfernt Schurken oder unberechtigte Nachrichten, bevor sie von einer empfangenden Anwendung verarbeitet werden.
- Prüft, ob Nachrichten während der Übertragung von Warteschlange in Warteschlange nicht geändert wurden.
- Schützt die Daten nicht nur, wenn sie über das Netz fließt, sondern auch, wenn sie in eine Warteschlange gestellt wird.
- Sichert die vorhandenen proprietären und vom Kunden geschriebenen Anwendungen für IBM MQ.
- V 9.2.0 z/OS Ab IBM MQ 9.1.3 kann der AMS-Schutz in IBM MQ for z/OS optional aus Nachrichten entfernt werden oder Nachrichten hinzugefügt werden, die im Netz übertragen werden. Dies wird als *MCA-Abfang zwischen Servern* bezeichnet.
- ALW Ab IBM MQ 9.1.4 und IBM MQ 9.1.0 Fix Pack 4 wurde dem IBM MQ-Bibliothekscode, der im Anwendungsprogramm des Kunden ausgeführt wird, eine Prüfung hinzugefügt. Die Prüfung wird an einem frühen Zeitpunkt in der Initialisierung ausgeführt, um den Wert der Umgebungsvariable *AMQ_AMS_FIPS_OFF* zu lesen. Wenn ein Wert festgelegt ist, wird der GSKit-Code in einem Nicht-FIPS-Modus in dieser Anwendung ausgeführt.

Für AMS verfügbare Datenschutzniveaus

Es gibt drei Qualitäten des Schutzes für Advanced Message Security, `Integrity` `Privacy`und `Confidentiality`.

Der `Integrity`Schutz wird durch digitales Signieren gewährleistet, das Gewissheit darüber gibt, wer die Nachricht erstellt hat und dass die Nachricht nicht geändert oder manipuliert wurde.

Der`Privacy` -Schutz wird durch eine Kombination aus digitaler Signatur und Verschlüsselung bereitgestellt. Die Verschlüsselung stellt sicher, dass die Nachrichtendaten nur für den vorgesehenen Empfänger oder Empfänger sichtbar sind. Selbst wenn nicht berechtigte Empfänger eine Kopie der verschlüsselten Nachrichtendaten erhalten, können sie die eigentlichen Nachrichtendaten nicht selbst anzeigen.

Der `Confidentiality` -Schutz wird durch die Verschlüsselung mit optionaler Schlüsselwiederverwendung gewährleistet.

Auswirkung auf die Leistung

AMS stellt mit einer Kombination aus symmetrischen und asymmetrischen Verschlüsselungsroutinen die digitale Unterzeichnung und Verschlüsselung bereit. Da symmetrische Schlüsseloperationen im Vergleich zu den CPU-intensiven asymmetrischen Schlüsseloperationen sehr schnell sind, kann sich dies auf die Kosten für den Schutz einer großen Anzahl von Nachrichten mit AMS auswirken.

Asymmetrische Verschlüsselungsroutinen

Wenn Sie beispielsweise eine signierte Nachricht einreihen, wird der Nachricht-Hash mit Hilfe einer asymmetrischen Schlüsseloperation signiert.

Wenn Sie eine signierte Nachricht erhalten, wird eine weitere asymmetrische Schlüsseloperation zur Überprüfung des signierten Hash-Schlüssels verwendet.

Aus diesem Grund sind pro Nachricht mindestens zwei asymmetrische Tastenoperationen erforderlich, um die Nachrichtendaten zu signieren und zu überprüfen.

Asymmetrische und symmetrische kryptografische Routinen

Beim Eingeben einer verschlüsselten Nachricht wird ein symmetrischer Schlüssel generiert und anschließend mit einer asymmetrischen Schlüsseloperation für jeden beabsichtigten Empfänger der Nachricht verschlüsselt.

Die Nachrichtendaten werden dann mit dem symmetrischen Schlüssel verschlüsselt. Beim Abrufen der verschlüsselten Nachricht muss der vorgesehene Empfänger eine asymmetrische Schlüsseloperation verwenden, um den symmetrischen Schlüssel zu erkennen, der für die Nachricht verwendet wird.

Alle drei Schutzqualitäten enthalten daher unterschiedliche Elemente der CPU-intensiven asymmetrischen Schlüsseloperationen, die sich erheblich auf die maximal erreichbare Nachrichtenübertragungsrate für Anwendungen auswirken, die Nachrichten einreihen und Nachrichten erhalten.

Die`Confidentiality` -Richtlinien ermöglichen jedoch die Wiederverwendung symmetrischer Schlüssel für eine Nachrichtenfolge. Durch die Wiederverwendung symmetrischer Schlüssel können mit `Confidentiality` -Richtlinien erhebliche CPU-Kosteneinsparungen erzielt werden. Diese Betriebsart verwendet weiterhin das Format PKCS#7 , um einen symmetrischen Verschlüsselungsschlüssel gemeinsam zu nutzen. Es gibt jedoch keine digitale Signatur, die einige der asymmetrischen Tastenoperationen pro Nachricht überflüssig macht. Der symmetrische Schlüssel muss immer noch mit asymmetrischen Schlüsseloperationen für jeden Empfänger verschlüsselt werden, aber der symmetrische Schlüssel kann optional über mehrere Nachrichten wiederverwendet werden, die für dieselben Empfänger bestimmt sind. Wenn die Schlüsselwiederverwendung nach Richtlinie zulässig ist, erfordert nur die erste Nachricht asymmetrische Schlüsseloperationen. Nachfolgende Nachrichten müssen nur symmetrische Tastenoperationen verwenden.

Schlüsselwiederverwendung

Mit `Confidentiality` -Richtlinien können Sie den Ansatz der symmetrischen Schlüsselwiederverwendung verwenden, um die Kosten für die Verschlüsselung einer Reihe von Nachrichten, die in dieselbe Warteschlange eingereiht werden und für denselben Empfänger bestimmt sind, erheblich zu reduzieren.

Wenn Sie beispielsweise 10 verschlüsselte Nachrichten in dieselbe Empfängergruppe einreihen, wird ein symmetrischer Schlüssel generiert und dann für die erste Nachricht verschlüsselt, wobei für jeden beabsichtigten Empfänger der Nachricht eine asymmetrische Schlüsseloperation verwendet wird.

Der verschlüsselte symmetrische Schlüssel kann dann basierend auf richtliniengesteuerten Begrenzungen wiederverwendet werden, indem nachfolgende Nachrichten für dieselben Empfänger verwendet werden. Damit der symmetrische Schlüssel von nachfolgenden Nachrichten wiederverwendet werden kann, muss die Anwendung die Warteschlange nach dem Einreihen einer Nachricht in die Warteschlange geöffnet lassen. Der symmetrische Schlüssel kann nicht von MQPUT1 -Operationen wiederverwendet werden. Eine Anwendung, die verschlüsselte Nachrichten erhält, kann die gleiche Optimierung anwenden, da die Anwendung erkennen kann, wenn sich ein symmetrischer Schlüssel nicht geändert hat, und den Aufwand für das Abrufen des symmetrischen Schlüssels zu vermeiden.

In diesem Beispiel können 90% der asymmetrischen Tastenoperationen sowohl durch das Einlegen als auch durch das Abrufen von Anwendungen durch die Verwendung desselben Schlüssels vermieden werden.

Weitere Informationen zur Verwendung der Schlüsselwiederverwendung finden Sie unter:

- MQSC-Befehl SET POLICY
- Steuerbefehl setmqspl
- IBM i IBM i-Befehl SETMQMSPL

Zentrale Konzepte in AMS

In diesem Abschnitt erhalten Sie Informationen zu den zentralen Konzepten in Advanced Message Security, um die Arbeitsweise der Tools besser zu verstehen und sie effektiv verwalten zu können.

Public Key Infrastructure und Advanced Message Security

Public Key Infrastructure (PKI) ist ein System von Einrichtungen, Richtlinien und Services, die die Verwendung öffentlicher Schlüsselverschlüsselungsfunktionen unterstützen, um eine sichere Kommunikation zu erhalten.

Es gibt keinen einzigen Standard, der die Komponenten einer öffentlichen Schlüsselinfrastruktur definiert, aber ein PKI umfasst normalerweise die Verwendung öffentlicher Schlüsselzertifikate und umfasst Zertifizierungsstellen (CA) und andere Registrierungsstellen (RA), die die folgenden Services bereitstellen:

- Digitale Zertifikate ausstellen
- Digitale Zertifikate validieren
- Digitale Zertifikate werden zurückgeschworen
- Zertifikate verteilen

Die Identität von Benutzern und Anwendungen wird durch das Feld **Definierter Name (DN)** in einem Zertifikat dargestellt, das signierten oder verschlüsselten Nachrichten zugeordnet ist. Advanced Message Security verwendet diese Identität, um einen Benutzer oder eine Anwendung darzustellen. Zur Authentifizierung dieser Identität muss der Benutzer oder die Anwendung Zugriff auf den Schlüsselspeicher haben, in dem das Zertifikat und der zugehörige private Schlüssel gespeichert sind. Jedes Zertifikat wird durch eine Bezeichnung im Keystore dargestellt.

Zugehörige Konzepte

„Keystores und Zertifikate mit AMS verwenden” auf Seite 682
Um für IBM MQ-Anwendungen einen transparenten Verschlüsselungsschutz bereitzustellen, verwendet Advanced Message Security die Schlüsselspeicherdatei, in der Zertifikate für öffentliche Schlüssel und private Schlüssel gespeichert werden. Unter z/OS wird anstelle einer Schlüsselspeicherdatei ein SAF-Schlüsselring verwendet.

Digitale Zertifikate in AMS

Advanced Message Security verknüpft Benutzer und Anwendungen mit digitalen X.509-Standardzertifikaten. X.509-Zertifikate werden in der Regel von einer anerkannten Zertifizierungsstelle (CA) signiert und beinhalten private und öffentliche Schlüssel, die für die Verschlüsselung und Entschlüsselung verwendet werden.

Digitale Zertifikate bieten Schutz vor der Imitation, indem sie einen öffentlichen Schlüssel an seinen Eigner binden, unabhängig davon, ob dieser Eigentümer eine Einzelperson, ein Warteschlangenmanager oder eine andere Entität ist. Digitale Zertifikate werden auch als öffentliche Schlüsselzertifikate bezeichnet, da sie Ihnen bei Verwendung eines asymmetrischen Schlüsselschemas die Gewissheit über das Eigentumsrecht an einem öffentlichen Schlüssel geben. Für dieses Schema ist es erforderlich, dass ein öffentlicher Schlüssel und ein privater Schlüssel für eine Anwendung generiert werden. Daten, die mit dem öffentlichen Schlüssel verschlüsselt werden, können nur mit Hilfe des entsprechenden privaten Schlüssels entschlüsselt werden, während Daten, die mit dem privaten Schlüssel verschlüsselt werden, nur mit dem entsprechenden öffentlichen Schlüssel entschlüsselt werden können. Der private Schlüssel wird in einer Schlüsseldatenbankdatei gespeichert, die kennwortgeschützt ist. Nur der zugehörige Eigner hat den Zugriff auf den privaten Schlüssel, der zum Entschlüsseln von Nachrichten verwendet wird, die mit dem entsprechenden öffentlichen Schlüssel verschlüsselt wurden.

Wenn öffentliche Schlüssel direkt von ihrem Eigner an eine andere Entität gesendet werden, besteht die Gefahr, dass die Nachricht abgefangen und der öffentliche Schlüssel durch einen anderen ersetzt wird. Dies wird auch als "Man-in-the-middle"-Angriff bezeichnet. Die Lösung besteht darin, die öffentlichen Schlüssel über eine vertrauenswürdige dritte Partei auszutauschen und dem Benutzer eine hohe Sicherheit zu geben, dass der öffentliche Schlüssel zu der Entität gehört, mit der Sie kommunizieren. Anstatt Ihren öffentlichen Schlüssel direkt zu senden, bitten Sie einen vertrauenswürdigen Dritten, ihn in ein digitales Zertifikat zu integrieren. Der anerkannte Dritte, der digitale Zertifikate ausgibt, wird als Zertifizierungsstelle (CA) bezeichnet.

Weitere Informationen zu digitalen Zertifikaten finden Sie unter What is in a digital certificate .

Ein digitales Zertifikat enthält den öffentlichen Schlüssel für eine Entität und gibt an, dass der öffentliche Schlüssel zu dieser Entität gehört:

- Wenn ein Zertifikat für eine einzelne Entität vorhanden ist, wird es als *persönliches Zertifikat* oder *Benutzerzertifikat* bezeichnet.
- Wenn ein Zertifikat für eine Zertifizierungsstelle ausgestellt wurde, wird das Zertifikat als *CA-Zertifikat* oder *Unterzeichnerzertifikat* bezeichnet.

Anmerkung: Advanced Message Security unterstützt selbst signierte Zertifikate in Java-Anwendungen und in nativen Anwendungen

Zugehörige Konzepte

„Kryptografie” auf Seite 10
Bei der Verschlüsselung handelt es sich um den Konvertierungsprozess zwischen lesbarem Text, dem so genannten *Klartext* , und einem nicht lesbaren Format mit dem Namen *chiffriktext* .

Multi *Objektberechtigungsmanager und AMS*

Auf Multiplatforms ist der Objektberechtigungsmanager (Object Authority Manager, OAM) die Berechtigungsservicekomponente, die mit den IBM MQ-Produkten bereitgestellt wird.

Der Zugriff auf Advanced Message Security-Entitäten wird über IBM MQ-Benutzergruppen und den OAM gesteuert. Administratoren können die Befehlszeilenschnittstelle verwenden, um Berechtigungen nach Bedarf zu erteilen oder zu widerrufen. Unterschiedliche Benutzergruppen können unterschiedliche Arten von Zugriffsberechtigungen für dieselben Objekte haben. Eine Gruppe kann z. B. sowohl PUT-als auch GET-Operationen für eine bestimmte Warteschlange ausführen, während eine andere Gruppe nur zum Durchsuchen der Warteschlange berechtigt ist. In ähnlicher Weise können einige Gruppen GET-und PUT-Berechtigungen für eine Warteschlange haben, aber sie dürfen die Warteschlange nicht ändern oder löschen.

Über den OAM können Sie Folgendes steuern:

- Zugriff auf Advanced Message Security -Objekte über Message Queue Interface (MQI). Wenn ein Anwendungsprogramm versucht, auf Objekte zuzugreifen, prüft der OAM, ob das Benutzerprofil, das die Anforderung stellt, die Berechtigung für die angeforderte Operation hat. Dies bedeutet, dass Warteschlangen und die Nachrichten in Warteschlangen vor unbefugtem Zugriff geschützt werden können.
- Berechtigung zum Verwenden von PCF-und MQSC-Befehlen.

Zugehörige Konzepte

Objektberechtigungsmanager

Message Queue Interface (MQI) - Übersicht

Von Advanced Message Security unterstützte Technologie

Advanced Message Security hängt von mehreren IT-Komponenten ab, mit denen eine Sicherheitsinfrastruktur bereitgestellt wird.

Advanced Message Security unterstützt die folgenden IBM MQ APIs (Application Programming Interfaces, Anwendungsprogrammierschnittstellen):

- Nachrichtenwarteschlangenschnittstelle (MQI)

- IBM MQ Java Message Service (JMS) 1.0.2 und 1.1.
- IBM MQ-Basisklassen für Java
- IBM MQ-Klasse für .Net in einem nicht verwalteten Modus

Anmerkung: Advanced Message Security unterstützt X.509-konforme Zertifizierungsstellen.

Bekannte Einschränkungen von AMS

Es gibt eine Reihe von IBM MQ-Optionen, die nicht unterstützt werden oder Einschränkungen für Advanced Message Security haben.

- Die folgenden IBM MQ-Optionen werden nicht unterstützt oder unterliegen Einschränkungen:

  Publish/Subscribe
  Einer der Hauptvorteile eines Publish/Subscribe-Messaging-Modells über Punkt-zu-Punkt-Verbindungen besteht darin, dass die sendenden und empfangenden Anwendungen keine Informationen über die zu sendenden und zu empfangenden Daten benötigen. Dieser Vorteil entfällt allerdings bei der Verwendung von Advanced Message Security-Richtlinien, die bestimmte Empfänger oder berechtigte Unterzeichner definieren müssen. Es ist möglich, dass eine Anwendung über eine Aliaswarteschlangendefinition, die durch eine Richtlinie geschützt ist, in einem Topic veröffentlicht wird. Es ist auch möglich, dass eine subskribierende Anwendung Nachrichten aus einer Richtlinie für geschützte Warteschlangen erhält. Es ist nicht möglich, eine Richtlinie direkt einer Themenzeichenfolge zuzuordnen, da die Richtlinien nur Warteschlangendefinitionen zugeordnet werden können.

  Kanaldatenkonvertierung
  Die geschützten Nutzdaten einer geschützten Advanced Message Security-Nachricht werden im Binärformat übertragen. Dadurch wird sichergestellt, dass die Datenkonvertierung in einem Kanal zwischen Anwendungen die Aufnahme von Nachrichten nicht inaktiviert. Anwendungen, die Nachrichten aus einer richtliniengeschützten Warteschlange abrufen, sollten die Datenkonvertierung anfordern, die Konvertierung der geschützten Nutzdaten wird versucht, nachdem Nachrichten erfolgreich überprüft und ungeschützt wurden.

  Verteilerlisten
  Advanced Message Security-Richtlinien können verwendet werden, wenn Anwendungen, die Nachrichten in Verteilerlisten stellen, geschützt werden, vorausgesetzt, für jede Zielwarteschlange in der Liste ist eine identische Richtlinie definiert. Wenn inkonsistente Richtlinien identifiziert werden, wenn eine Anwendung eine Verteilerliste öffnet, schlägt die Operation zum Öffnen fehl und es wird ein Sicherheitsfehler an die Anwendung zurückgegeben.

  Anwendungsnachrichtensegmentierung
  Die Größe der richtliniengeschützten Nachrichten wird erhöht, und es ist nicht möglich, dass Anwendungen die Segmentgrenzen einer Nachricht genau angeben.

  Anwendungen, die IBM MQ classes for .NET in einem verwalteten Modus verwenden (Clientverbindungen)
  Anwendungen, die IBM MQ classes for .NET in einem verwalteten Modus (Clientverbindungen) verwenden, werden nicht unterstützt.

  Anmerkung: MCA-Abfangprozesse können verwendet werden, um nicht unterstützte Clients zu ermöglichen, AMSzu verwenden.

  Message Service Client für .NET-Anwendungen (XMS) in einem verwalteten Modus
  Der Message Service-Client für .NET-Anwendungen (XMS) in einem verwalteten Modus wird nicht unterstützt.

  Anmerkung: MCA-Abfang kann verwendet werden, um nicht unterstützte Clients zu ermöglichen, AMS zu verwenden.

  IBM MQ-Warteschlangen, die von der IMS-Bridge verarbeitet werden
  IBM MQ-Warteschlangen, die von der IMS-Bridge verarbeitet werden, werden nicht unterstützt.

  Anmerkung: AMS wird in CICS-Brückenwarteschlangen unterstützt. Es sollte dieselbe Benutzer-ID für MQPUT (verschlüsseln) und MQGET (entschlüsseln) in CICS-Brückenwarteschlangen verwendet werden.

Abruffunktion 'Put to waiting getter'
Das Einreihen in wartende Getter wird für Getter-Anwendungen für Warteschlangen mit definierten AMS-Richtlinien nicht unterstützt.

V 9.2.0 **MCA-Abfangprozess zwischen Servern**
Von IBM MQ for z/OS 9.1.3 wird die Server-zu-Server-MCA-Abfangfunktion nur für Sender-, Server-, Empfänger- und Anforderkanaltypen unterstützt.

- Benutzer sollten vermeiden, dass mehr als ein Zertifikat mit demselben definierten Namen in einer einzigen Keystore-Datei vorhanden ist, da die Auswahl des Zertifikats, das beim Schutz einer Nachricht verwendet werden soll, nicht definiert ist.
- AMS wird in JMS nicht unterstützt, wenn die Eigenschaft **WMQ_PROVIDER_VERSION** auf 6 gesetzt ist.
- Der AMS-Abfangprozess wird für AMQP-oder MQTT-Kanäle nicht unterstützt.

V 9.2.0 z/OS Advanced Message Security-Abfangprozesse in Nachrichtenkanälen

Unter z/OSstellt das Abfangen von Advanced Message Security (AMS) eine zusätzliche Option des Sicherheitsrichtlinienschutzes (SPLPROT) für Sender-, Server-, Empfänger-und Requesterkanäle bereit, mit der Sie AMS unterstützen und mit Geschäftspartnern kommunizieren können, die AMSnicht unterstützen.

Im Beispiel eines Clearing House, das mit einer Bank kommuniziert, wird in Abbildung 1 gezeigt, dass AMS ohne den AMS-Abfangprozess von beiden Seiten des Systems unterstützt werden muss.

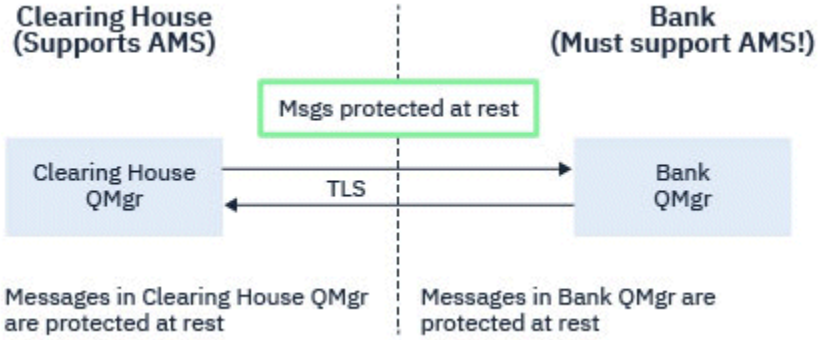


Abbildung 32. Verwendung von AMS ohne AMS-Abfangprozess

Wenn beispielsweise in Ihrem Unternehmen AMS konfiguriert ist, aber nicht alle Geschäftspartner AMS unterstützen, können Sie den Schutz von ausgehenden Nachrichten entfernen und eingehende Nachrichten in Kanälen an und von diesen Geschäftspartnern, die AMS nicht unterstützen, schützen, was einen wesentlichen Vorteil der Option mit dem AMS-Abfangprozess darstellt.

Dieses Szenario wird anhand des Beispiels eines Clearing House und Banken in Abbildung 2 dargestellt. Dabei besteht ein Nachrichtenfluss zwischen Clearing House, Banken und Geschäftspartnern, bei dem einige Institutionen AMS haben und andere nicht.

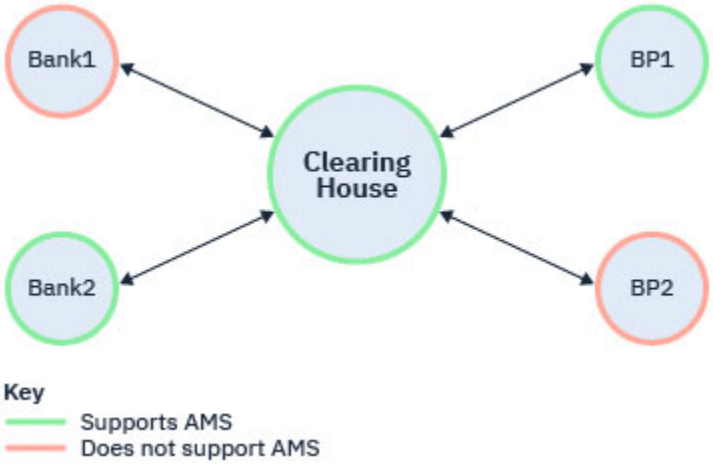


Abbildung 33. Einige Partner unterstützen AMS und andere nicht

In der Regel sind die Kanäle TLS-fähig.

Es kann jedoch den Fall geben, in dem einige Banken und Geschäftspartner AMS nicht unterstützen, und es erforderlich ist, Nachrichten zwischen allen Banken und Geschäftspartnern austauschen zu können. Dieses Szenario wird in Abbildung 3 dargestellt.

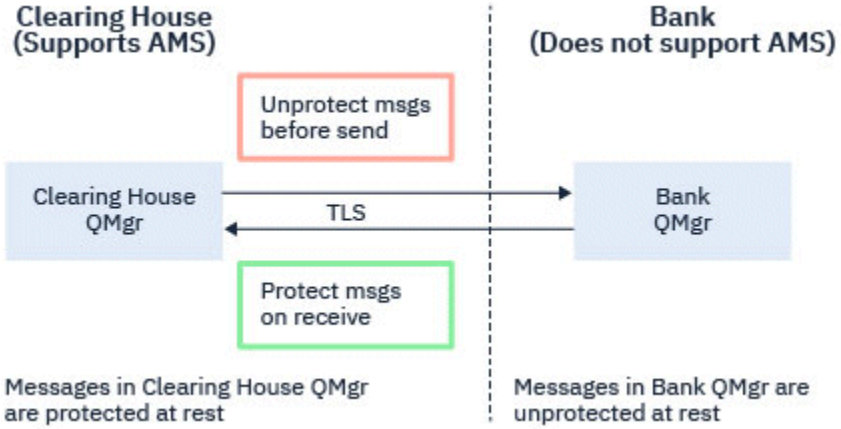


Abbildung 34. Nachrichtenfluss zwischen Geschäftspartnern

Zugehörige Tasks

Beispielkonfigurationen für die Kanalabfangprozedur des Nachrichtenaustausches zwischen Servern

V 9.2.0 z/OS *AMS - Abfangen auf Nachrichtenkanälen zwischen Servern*

Die Server-zu-Server-Nachrichtenkanal-Abfangung bietet eine Möglichkeit zur Steuerung, wenn Nachrichten über alle anwendbaren Advanced Message Security (AMS)-Richtlinien angewendet werden sollen, wenn Nachrichtenkanalagenten des Absenders Nachrichten von Übertragungswarteschlangen abrufen und Nachrichtenkanalagenten vom Empfängertyp Nachrichten an Zielwarteschlangen stellen.

Somit kann der AMS-Schutz bei der Kommunikation auf einem Warteschlangenmanager aktiviert werden mit Kanälen für Nachrichtenaustausch zwischen Servern des Typs "Sender", "Server", "Empfänger" und "Anforderer" und mit einem Warteschlangenmanager, bei dem AMS nicht aktiviert ist.

Das bedeutet, der Schutz von AMS-geschützten Nachrichten in AMS-fähigen Warteschlangenmanagern kann aufgehoben werden, bevor ein Senden an Warteschlangenmanager, die nicht AMS-fähig sind, erfolgt, und nicht geschützte Nachrichten, die von nicht-AMS-fähigen Warteschlangenmanagern empfangen werden, können von gültigen AMS-Richtlinien auf AMS-fähigen Warteschlangenmanagern geschützt werden.

Abfangen des Kanals für Nachrichtenaustausch zwischen Servern konfigurieren

Das Abfangen des Kanals für Nachrichtenaustausch zwischen Servern wird mit dem SPLPROT-Attribut auf Kanälen mit dem Kanaltyp "Sender", "Server", "Empfänger" oder "Anforderer" konfiguriert. Die verfügbaren Optionen, um das Verhalten zu konfigurieren, sind abhängig vom angegebenen Kanaltyp:

PASSTHRU
Alle vom Nachrichtenkanalagenten für diesen Kanal gesendeten oder empfangenen Nachrichten werden unverändert durchgeleitet.

Dieser Wert gilt für Kanäle des Kanaltyps (**CHLTYPE**) SDR, SVR, RCVR oder RQSTR und er ist der Standardwert.

REMOVE
Der AMS-Schutz wird aus Nachrichten, die vom Nachrichtenkanalagenten aus der Übertragungswarteschlange abgerufen werden, entfernt und die Nachrichten werden an den Partner gesendet.

Wenn der Nachrichtenkanalagent eine Nachricht aus der Übertragungswarteschlange abruft und eine AMS-Richtlinie für die Übertragungswarteschlange definiert ist, wird die Richtlinie angewendet, um einen vorhandenen AMS-Schutz vor dem Senden der Nachricht über den Kanal aus der Nachricht zu entfernen. Wenn keine AMS-Richtlinie für die Übertragungswarteschlange definiert ist, wird die Nachricht unverändert gesendet.

Dieser Wert ist nur für Kanäle mit dem Kanaltyp SDR oder SVR gültig.

ASPOLICY
Auf Basis der für die Zielwarteschlange definierten Richtlinie wird der AMS-Schutz auf eingehende Nachrichten angewendet, bevor sie in die Zielwarteschlange gestellt werden.

Wenn der Nachrichtenkanalagent eine eingehende Nachricht empfängt und eine AMS-Richtlinie für die Zielwarteschlange definiert ist, wird der AMS-Schutz auf die Nachricht angewendet, bevor sie in die Zielwarteschlange eingereiht wird. Wenn keine AMS-Richtlinie für die Zielwarteschlange definiert ist, wird die Nachricht unverändert in die Zielwarteschlange eingereiht.

Dieser Wert ist nur für Kanäle mit dem Kanaltyp RCVR oder RQSTR gültig.

Benutzer-ID für Nachrichtenkanalabfangfunktion

Die Anforderungen für Benutzer-IDs, die bei der Kanalabfangfunktion für den Nachrichtenaustausch zwischen Servern verwendet werden, sind die gleichen wie diejenigen für bestehende AMS-fähige Anwendungen. Für einen aktiven Kanal erhält der sendende Nachrichtenkanalagent Nachrichten aus einer Übertragungswarteschlange, und der empfangende Nachrichtenkanalagent reiht Nachrichten in Zielwarteschlangen ein. Das Feld für die Benutzer-ID des Nachrichtenkanalagenten (MCAUSER), das auf Kanälen für den Nachrichtenaustausch zwischen Servern festgelegt ist, definiert die Benutzer-ID, unter der Nachrichtenkanalagenten put- und get-Anforderungen ausführen.

Bei der Kanalabfangfunktion für den Nachrichtenaustausch zwischen Servern werden die AMS-Funktionen während der get- und put-Anforderungen ausgeführt, wie bei anderen AMS-fähigen Anwendungen. Daher haben die Benutzer-IDs des Nachrichtenkanalagenten dieselben Anforderungen wie die für AMS-Anwendungsbenutzer-IDs.

Der MCAUSER, der für die Ausführung des put- und get-Befehls verwendet wird, ist konfigurierbar und davon abhängig, ob es sich um einen abgehenden oder eingehenden Kanal handelt. Im Abschnitt MCAUSER finden Sie Informationen dazu, wie die ausgewählte Benutzer-ID Aktionen für den Nachrichtenkanalagenten ausführt. Als solche ist die Benutzer-ID, unter der der Kanalinitiator ausgeführt wird, die Benutzer-ID, die für AMS-Funktionen verwendet werden soll, die während der Kanalabfragefunktion für den Nachrichtenaustausch zwischen Servern ausgeführt werden. Daher haben die Benutzer-IDs dieselben Anforderungen wie die für AMS-Anwendungsbenutzer-IDs.

Die Authentifizierung wird unter Verwendung der vorhandenen Regeln für den Kanal ausgeführt, die für Kanäle mit PUTAUT-Konfiguration detailliert beschrieben werden. Weitere Informationen finden Sie unter Vom Kanalinitiator verwendete Benutzer-IDs.

Anmerkung: Die Kanalabfangfunktion für den Nachrichtenaustausch zwischen Servern berücksichtigt nicht den Wert des Kanalattributs PUTAUT.

Nachrichtengröße und MAXMSGL

Auf Grund des AMS-Schutzes übertrifft die Größe von geschützten Nachrichten die der ursprünglichen Nachrichten.

Geschützte Nachrichten sind größer als ungeschützte. Daher muss der Wert des Attributs **MAXMSGL** sowohl bei Warteschlangen als auch bei Kanälen möglicherweise geändert werden, um die Größe der geschützten Nachrichten zu berücksichtigen.

Zugehörige Verweise

Beispielkonfigurationen für die Kanalabfangprozedur des Nachrichtenaustausches zwischen Servern

Fehlerbehandlung für AMS

IBM MQ Advanced Message Security definiert eine Fehlerbehandlungswarteschlange für die Verwaltung von Nachrichten mit Fehlern oder Nachrichten, die nicht ungeschützt sein können.

Fehlerhafte Nachrichten werden als Ausnahmefälle behandelt. Wenn eine empfangene Nachricht die Sicherheitsanforderungen für die Warteschlange nicht erfüllt, z. B., wenn die Nachricht signiert wird, wenn sie verschlüsselt werden soll, oder die Entschlüsselung oder die Signaturprüfung fehlschlägt, wird die Nachricht an die Fehlerbehandlungswarteschlange gesendet. Eine Nachricht kann aus den folgenden Gründen an die Fehlerbehandlungswarteschlange gesendet werden:

- Datenschutzniveau-Es besteht eine Diskrepanz zwischen der empfangenen Nachricht und der QOP-Definition in der Sicherheitsrichtlinie, die eine Diskrepanz zwischen den empfangenen Nachrichten und der QOP-Definition hat.
- Entschlüsselungsfehler-die Nachricht kann nicht entschlüsselt werden.
- PDMQ-Header-Fehler - Auf den Nachrichtenheader von Advanced Message Security (AMS) kann nicht zugegriffen werden.
- Größenabweichung-die Länge einer Nachricht nach der Entschlüsselung ist anders als erwartet.
- Verschlüsselungsalgorithmusstärke stimmen nicht überein-der Algorithmus für die Nachrichtenverschlüsselung ist schwächer als erforderlich.
- Unbekannter Fehler-unerwarteter Fehler aufgetreten.

AMS verwendet das SYSTEM.PROTECTION.ERROR.QUEUE als Fehlerbehandlungswarteschlange. Alle Nachrichten, die von IBM MQ AMS in SYSTEM.PROTECTION.ERROR.QUEUE wird ein MQDLH-Header vorangestellt.

Ihr IBM MQ -Administrator kann auch das SYSTEM.PROTECTION.ERROR.QUEUE als Aliaswarteschlange, die auf eine andere Warteschlange verweist.

V 9.2.0 z/OS Ab IBM MQ 9.1.3 gilt unter IBM MQ for z/OS bei der Verwendung der Überwachung von Nachrichtenkanalagenten zwischen Servern Folgendes:

- Wenn Nachrichten in IBM MQ AMS aus einer der zuvor genannten Ursachen aus der Übertragungswarteschlange in die Fehlerbehandlungswarteschlange verschoben werden, fährt der Sender-MCA einfach mit der Verarbeitung der nächsten verfügbaren Nachricht in der Übertragungswarteschlange fort.
- Allgemein gelten vorhandene Kanalregeln für folgende Aktionen:
  - Einreihen von Nachrichten in die Warteschlange für nicht zustellbare Nachrichten, und
  - Aktionen, die vorgenommen werden, wenn das Einreihen von Nachrichten in die Warteschlange für nicht zustellbare Nachrichten fehlschlägt.

Weitere Informationen zu bestimmten Szenarios finden Sie unter „Nicht zugestellte Nachrichten für AMS unter z/OS” auf Seite 644.

V 9.2.0 z/OS *Nicht zugestellte Nachrichten für AMS unter z/OS*

In diesem Abschnitt werden bestimmte Szenarios beschrieben, die sich auf die Überwachung von Nachrichtenkanalagenten (Message Channel Agent, MCA) zwischen Servern unter IBM MQ for z/OS beziehen.

Ab IBM MQ 9.1.3 gilt unter IBM MQ for z/OS bei der Verwendung der Überwachung von Nachrichtenkanalagenten zwischen Servern Folgendes:

- Wenn nach dem Erhalten einer Nachricht und dem Aufheben des Schutzes der Sencer-MCA eine Nachricht aus irgendeinem Grund nicht zustellt (z. B. weil die Nachricht für den Kanal zu groß ist), verschiebt der Sencer-MCA, wenn für das USEDLQ-Senderkanal-Attribut YES festgelegt wurde, die Nachricht zur lokalen Warteschlange für nicht zustellbare Nachrichten (DLQ).

  Wenn SYSTEM.DEAD.LETTER.QUEUE als lokale Warteschlange für nicht zustellbare Nachrichten verwendet wird, wird die Nachricht ungeschützt platziert.

  Anmerkung: IBM MQ AMS unterstützt nicht den Schutz von Nachrichten, die in Systemwarteschlangen eingereiht sind.

  Wenn eine angegebene DLQ als lokale DLQ verwendet wird, werden die Nachrichten geschützt eingereiht, wenn Sie eine IBM MQ AMS-Richtlinie mit dem gleichen Namen wie die angegebene DLQ definiert haben, und sie werden ungeschützt eingereiht, wenn keine geeignete Richtlinie definiert ist.
- Wenn eine Nachricht aus einem beliebigen Grund nicht in die lokale DLQ eingereiht werden kann und NPMSPEED für den Kanal auf NORMAL gesetzt ist oder es sich um eine persistente Nachricht handelt, wird der aktuelle Nachrichtenstapel zurückgestellt und der Kanal wird in den Status RETRY versetzt. Andernfalls wird die Nachricht gelöscht und der sendende Nachrichtenkanalagent fährt mit der Verarbeitung der nächsten Nachricht in der Übertragungsschlange fort.
- Da die Sicherheitsrichtlinien keine Auswirkungen auf SYSTEM.DEAD.LETTER.QUEUE oder die anderen in „Schutz der Systemwarteschlange in AMS” auf Seite 721 aufgeführten Systemwarteschlangen haben, werden bei Verwendung von SYSTEM.DEAD.LETTER.QUEUE die von den Nachrichtenkanalagenten in diese Warteschlange eingereihten Nachrichten unverändert platziert. Wenn also Nachrichten zuvor geschützt waren, werden sie auch geschützt platziert; andernfalls werden sie ungeschützt platziert.

  Wenn das Attribut DEADQ des Warteschlangenmanagers auf den Namen einer alternativen Warteschlange (keine Systemwarteschlange) für nicht zustellbare Nachrichten gesetzt ist und keine AMS-Richtlinie mit dem gleichen Namen vorhanden ist, werden die von den Nachrichtenkanalagenten in diese Warteschlange eingereihten Nachrichten unverändert platziert. Wenn also Nachrichten zuvor geschützt waren, werden sie auch geschützt platziert; andernfalls werden sie ungeschützt platziert.

  Wenn das Attribut DEADQ des Warteschlangenmanagers auf den Namen einer alternativen Warteschlange (keine Systemwarteschlange)für nicht zustellbare Nachrichten gesetzt ist und eine AMS-Richtlinie mit dem gleichen Namen wie die DLQ vorhanden ist, werden die von den Nachrichtenkanalagenten in diese Warteschlange eingereihten Nachrichten mit dieser Richtlinie geschützt. Wenn die Nachricht bereits geschützt war, wird sie nicht erneut geschützt; dadurch soll ein doppelter Schutz vermieden werden. Wenn keine AMS-Richtlinie mit dem gleichen Namen vorhanden ist, werden Nachrichten unverändert platziert.
- Wenn eine Richtlinie für die DLQ mit einer inaktiven tolerate-Option im Befehl setmqspl vorhanden ist, d. h. '**-t** O', schlägt das Einreihen in die DLQ fehl, wenn die Nachricht nicht AMS-geschützt ist, und weist daher keinen PDMQ-Header auf. Dies geschieht, wenn der Empfänger die Nachricht ohne PDMQ-Header erhält. Das bedeutet, dass die Komponente, mit der die Nachricht ursprünglich eingereiht wurde, über keine Richtlinie für das Ziel verfügte und im Empfänger SPLPROT(ASPOLICY) nicht festgelegt ist.
- Ein MCA kann möglicherweise eine Nachricht nicht in die DLQ einreihen, wenn die AMS-Richtlinie, die für die DLQ definiert wurde, der Benutzer-ID, unter der der Kanalinitiator ausgeführt wird, nicht erlaubt, die Nachricht zu schützen.
- Empfängerkanäle reihen in der Regel nicht zugestellte Nachrichten in die lokale DLQ ein, während Senderkanäle Nachrichten, die aus einem beliebigen Grund nicht verarbeitet werden können, beispielsweise, weil sie zu groß für die Warteschlange sind oder über einen fehlerhaften MQXQH-Header verfügen, normalerweise in der lokalen DLQ platzieren.
- DLQ-Handler berücksichtigen in der Regeln nur den DLQ-Header (DLH) und nicht die Nachrichtennutzdaten selbst. Deshalb ermitteln Handler auch dann den Grund, warum die Nachricht in die DLQ eingereiht wurde, wenn die Nachrichtennutzdaten geschützt sind.
- Wenn eine DLQ nicht definiert ist, wird der Kanal

– abnormal beendet (und wechselt in einen Wiederholungsstatus), wenn eine persistente Nachricht nicht übergeben werden kann.
– eine nicht persistente, nicht zugestellte Nachricht löschen und die Ausführung fortsetzten.

Zugehörige Konzepte

„Fehlerbehandlung für AMS” auf Seite 644
IBM MQ Advanced Message Security definiert eine Fehlerbehandlungswarteschlange für die Verwaltung von Nachrichten mit Fehlern oder Nachrichten, die nicht ungeschützt sein können.

Benutzerszenarien für AMS

Machen Sie sich mit möglichen Szenarios vertraut, um die Geschäftsziele zu verstehen, die Sie mit Advanced Message Security erreichen können.

Windows *Schnelleinstieg für AMS auf Windows-Plattformen*

Verwenden Sie dieses Handbuch für die schnelle Konfiguration von Advanced Message Security, um die Nachrichtensicherheit auf Windows-Plattformen bereitzustellen. Nach Abschluss der Ausführung haben Sie eine Schlüsseldatenbank erstellt, um die Benutzeridentitäten und die definierten Richtlinien für die Signierung/Verschlüsselung für Ihren Warteschlangenmanager zu überprüfen.

Vorbereitende Schritte

Sie sollten mindestens die folgenden Features auf Ihrem System installiert haben:

- Server
- Development Toolkit (für die Beispielprogramme)
- Erweiterte Nachrichtensicherheit

Einzelheiten finden Sie unter IBM MQ-Funktionen für Windows-Systeme.

Informationen zur Verwendung des Befehls **setmqenv** zum Initialisieren der aktuellen Umgebung, damit die entsprechenden IBM MQ -Befehle vom Betriebssystem lokalisiert und ausgeführt werden können, finden Sie im Abschnitt setmqenv (set IBM MQ environment).

1. WS-Manager und eine Warteschlange erstellen

Informationen zu diesem Vorgang

In allen folgenden Beispielen wird eine Warteschlange mit dem Namen `TEST.Q` verwendet, um Nachrichten zwischen Anwendungen zu übergeben. Advanced Message Security verwendet Interceptors, um Nachrichten an dem Punkt zu signieren und zu verschlüsseln, an dem Sie in der IBM MQ-Infrastruktur über die Standardschnittstelle von IBM MQ eintreffen. Die Basiseinrichtung wird in IBM MQ vorgenommen und in den folgenden Schritten konfiguriert.

Sie können IBM MQ Explorer verwenden, um den Warteschlangenmanager QM_VERIFY_AMS und seine lokale Warteschlange mit dem Namen `TEST.Q` zu erstellen, indem Sie alle Standardeinstellungen des Assistenten verwenden, oder Sie können die Befehle in `C:\Programme\IBM\MQ\bin`verwenden. Denken Sie daran, dass Sie ein Mitglied der `mqm` -Benutzergruppe sein müssen, um die folgenden Verwaltungsbefehle auszuführen.

Vorgehensweise

1. Einen WS-Manager erstellen

```
crtmqm QM_VERIFY_AMS
```

2. WS-Manager starten

```
strmqm QM_VERIFY_AMS
```

3. Erstellen Sie eine Warteschlange mit dem Namen TEST.Q , indem Sie den folgenden Befehl in **runmqsc** für Warteschlangenmanager QM_VERIFY_AMS eingeben.

```
DEFINE QLOCAL(TEST.Q)
```

Ergebnisse

Wenn die Prozedur abgeschlossen ist, zeigt der in **runmqsc** eingegebene Befehl Details zu TEST.Qan:

```
DISPLAY Q(TEST.Q)
```

2. Benutzer erstellen und berechtigen

Informationen zu diesem Vorgang

In diesem Beispiel werden zwei Benutzer angezeigt: alice, der Sender und bob, der Empfänger. Um die Anwendungswarteschlange verwenden zu können, müssen diese Benutzer berechtigt sein, sie zu verwenden. Außerdem müssen Sie die Zugriffsschutzrichtlinien, die wir definieren, erfolgreich verwenden, um Zugriff auf einige Systemwarteschlangen zu erhalten. Weitere Informationen zum Befehl **setmqaut** finden Sie unter **setmqaut**.

Vorgehensweise

1. Erstellen Sie die beiden Benutzer und stellen Sie sicher, dass HOMEPATH und HOMEDRIVE für diese beiden Benutzer festgelegt sind.
2. Benutzer berechtigen, eine Verbindung zum WS-Manager herzustellen und mit der Warteschlange zu arbeiten

```
setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get
```

3. Außerdem sollten Sie den beiden Benutzern die Möglichkeit geben, die Systemrichtlinienwarteschlange zu durchsuchen und Nachrichten in die Fehlerwarteschlange einzureihen.

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p alice -p bob +browse
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p alice -p bob +put
```

Achtung: IBM MQ optimiert die Leistung durch das Caching von Richtlinien, sodass Sie keine Datensätze nach Richtliniendetails auf dem SYSTEM.PROTECTION.POLICY.QUEUE in allen Fällen.

Von IBM MQ werden nicht alle verfügbaren Richtlinien zwischengespeichert. Wenn eine hohe Anzahl an Richtlinien vorhanden ist, wird von IBM MQ eine begrenzte Anzahl von Richtlinien zwischengespeichert. Wenn also der Warteschlangenmanager eine geringe Anzahl von Richtlinien definiert hat, ist es nicht erforderlich, die Option zum Durchsuchen für SYSTEM.PROTECTION.POLICY.QUEUE bereitzustellen.

Sie sollten jedoch die Berechtigung zum Durchsuchen dieser Warteschlange erteilen, falls eine hohe Anzahl an Richtlinien definiert ist, oder wenn Sie alte Clients verwenden. Die Warteschlange SYSTEM.PROTECTION.ERROR.QUEUE wird zum Einreihen von Fehlernachrichten verwendet, die vom AMS-Code generiert werden. Die Einreihungsberechtigung für diese Warteschlange wird nur überprüft, wenn Sie versuchen, eine Fehlernachricht in die Warteschlange einzureihen. Ihre Einreihungsberechtigung für die Warteschlange wird nicht überprüft, wenn Sie versuchen, eine Nachricht in eine AMS-geschützte Warteschlange einzureihen oder daraus abzurufen.

Ergebnisse

Die Benutzer werden jetzt erstellt, und die erforderlichen Berechtigungen werden ihnen erteilt.

Nächste Schritte

Um zu überprüfen, ob die Schritte korrekt ausgeführt wurden, verwenden Sie die Beispiele `amqsput` und `amqsget` wie in Abschnitt „7. Setup testen” auf Seite 651 beschrieben.

3. Schlüsseldatenbank und Zertifikate erstellen

Informationen zu diesem Vorgang

Der Interceptor benötigt den öffentlichen Schlüssel des sendenden Benutzers, um die Nachricht zu verschlüsseln. Daher muss die Schlüsseldatenbank der Benutzeridentitäten, die öffentlichen und privaten Schlüsseln zugeordnet sind, erstellt werden. Im realen System, in dem Benutzer und Anwendungen auf mehreren Computern verteilt sind, hat jeder Benutzer seinen eigenen privaten Schlüsselspeicher. Ebenso erstellen wir in diesem Handbuch Schlüsseldatenbanken für `alice` und bob und nutzen die Benutzerzertifikate zwischen ihnen gemeinsam.

Anmerkung: In diesem Handbuch verwenden wir Musteranwendungen, die in C geschrieben sind und die lokale Bindungen verwenden. Wenn Sie Java-Anwendungen mit Clientbindungen verwenden möchten, müssen Sie einen JKS-Keystore und Zertifikate mit dem Befehl **`keytool`** erstellen, der Teil der JRE ist (weitere Informationen finden Sie unter „Leitfaden für den Schnelleinstieg für AMS mit Java-Clients” auf Seite 669). Für alle anderen Sprachen und für Java-Anwendungen, die lokale Bindungen verwenden, sind die Schritte in diesem Handbuch korrekt.

Vorgehensweise

1. Erstellen Sie mit der grafischen Benutzerschnittstelle für IBM Key Management (`strmqikm.exe`) eine neue Schlüsseldatenbank für den Benutzer `alice`.

   ```
   Type:       CMS
   Filename:   alicekey.kdb
   Location:   C:/Documents and Settings/alice/AMS
   ```

   Anmerkung:

   - Es ist ratsam, ein sicheres Kennwort zu verwenden, um die Datenbank zu sichern.
   - Stellen Sie sicher, dass das Kontrollkästchen **Stashkennwort in eine Datei** ausgewählt ist.
2. Ändern Sie die Inhaltsansicht der Schlüsseldatenbank in **Personal Certificates** (Persönliche Zertifikate).
3. Wählen Sie **Neu selbst signiert** aus; selbst signierte Zertifikate werden in diesem Szenario verwendet.
4. Erstellen Sie mit den folgenden Feldern ein Zertifikat, das den Benutzer `alice` für die Verwendung in der Verschlüsselung identifiziert:

   ```
   Key label: Alice_Cert
   Common Name: alice
   Organisation: IBM
   Country: GB
   ```

   Anmerkung:

   - Im Sinne dieses Leitfadens verwenden wir selbst signiertes Zertifikat, das ohne Verwendung einer Zertifizierungsstelle erstellt werden kann. Bei Produktionssystemen ist es ratsam, keine selbst signierten Zertifikate zu verwenden, sondern sich auf Zertifikate zu stützen, die von einer Zertifizierungsstelle signiert wurden.
   - Der Parameter **`Key label`** gibt den Namen für das Zertifikat an, in dem die Interceptors nach den erforderlichen Informationen suchen.
   - Die Parameter **`Common Name`** und optionale Parameter geben die Details des **definierten Namens** (DN) an, der für jeden Benutzer eindeutig sein muss.
5. Wiederholen Sie die Schritte 1 bis 4 für Benutzer bob

Ergebnisse

Die beiden Benutzer `alice` und bob verfügen nun jeweils über ein selbst signiertes Zertifikat.

4. Keystore.conf erstellen

Informationen zu diesem Vorgang

Advanced Message Security-Interceptors müssen auf das Verzeichnis verweisen, in dem sich die Schlüsseldatenbanken und Zertifikate befinden. Dies erfolgt über die Datei `keystore.conf`, die diese Informationen im Klartextformat enthält. Jeder Benutzer muss über eine separate `keystore.conf`-Datei im Ordner `.mqs` verfügen. Dieser Schritt muss für `alice` und bobausgeführt werden.

Der Inhalt von `keystore.conf` muss das folgende Format haben:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

Beispiel

Für dieses Szenario wird der Inhalt des `keystore.conf` wie folgt aussehen:

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

Anmerkung:

- Der Pfad zu der Keystore-Datei muss ohne Dateierweiterung angegeben werden.
- Der Zertifikatskennsatz kann Leerzeichen enthalten, so zum Beispiel "Alice_Cert" und "Alice_Cert" (mit einem Leerzeichen am Ende), z.B. als Kennsätze von zwei unterschiedlichen Zertifikaten erkannt. Um Unklarheiten zu vermeiden, ist es jedoch besser, keine Leerzeichen im Namen der Bezeichnung zu verwenden.
- Es gibt die folgenden Keystore-Formate: CMS (Cryptographic Message Syntax), JKS (Java Keystore) und JCEKS (Java Cryptographic Extension Keystore). Weitere Informationen hierzu finden Sie unter „Struktur der Keystore-Konfigurationsdatei (keystore.conf) für AMS” auf Seite 684.
- *%HOMEDRIVE%\%HOMEPATH%*\`.mqs\keystore.conf` (z. B. C:\Documents and Settings\alice\.mqs\keystore.conf) ist die Standardposition, an der Advanced Message Security nach der `keystore.conf`-Datei sucht. Informationen zur Verwendung einer nicht standardmäßigen Position für die `keystore.conf` finden Sie unter „Keystores und Zertifikate mit AMS verwenden” auf Seite 682.
- Um das Verzeichnis `.mqs` zu erstellen, müssen Sie die Eingabeaufforderung verwenden.

5. Zertifikate gemeinsam nutzen

Informationen zu diesem Vorgang

Geben Sie die Zertifikate zwischen den beiden Schlüsseldatenbanken frei, so dass jeder Benutzer die andere identifizieren kann. Dazu wird jedes öffentliche Zertifikat jedes Benutzers in eine Datei extrahiert, die dann zur Schlüsseldatenbank des anderen Benutzers hinzugefügt wird.

Anmerkung: Gehen Sie vorsichtig vor, um die Option *extract* zu verwenden, und nicht die Option *export* . *Extrahieren* ruft den öffentlichen Schlüssel des Benutzers ab, während *export* sowohl den öffentlichen als auch den privaten Schlüssel erhält. Wenn Sie *export* versehentlich verwenden, würde Ihre Anwendung vollständig durch die Weitergabe des privaten Schlüssels beeinträchtigt.

Vorgehensweise

1. Extrahieren Sie das Zertifikat, das `alice` identifiziert, in eine externe Datei:

```
runmqakm -cert -extract -db "C:/Documents and Settings/alice/AMS/alicekey.kdb" -pw passw0rd
-label Alice_Cert -target alice_public.arm
```

2. Fügen Sie das Zertifikat dem bob ' s -Keystore hinzu:

```
runmqakm -cert -add -db "C:/Documents and Settings/bob/AMS/bobkey.kdb" -pw passw0rd -label
Alice_Cert -file alice_public.arm
```

3. Wiederholen Sie die Schritte für bob:

```
runmqakm -cert -extract -db "C:/Documents and Settings/bob/AMS/bobkey.kdb" -pw passw0rd -la
bel Bob_Cert -target bob_public.arm

runmqakm -cert -add -db "C:/Documents and Settings/alice/AMS/alicekey.kdb" -pw passw0rd -la
bel Bob_Cert -file bob_public.arm
```

Ergebnisse

Die beiden Benutzer `alice` und bob sind jetzt in der Lage, einander erfolgreich zu identifizieren, wenn sie selbst signierte und gemeinsam signierte Zertifikate erstellt haben.

Nächste Schritte

Überprüfen Sie, ob ein Zertifikat im Keystore vorhanden ist, indem Sie es mit der grafischen Benutzerschnittstelle durchsuchen oder die folgenden Befehle ausführen, um die Details zu drucken:

```
runmqakm -cert -details -db "C:/Documents and Settings/bob/AMS/bobkey.kdb" -pw passw0rd -label
Alice_Cert
```

```
runmqakm -cert -details -db "C:/Documents and Settings/alice/AMS/alicekey.kdb" -pw passw0rd -la
bel Bob_Cert
```

6. Warteschlangenrichtlinie definieren

Informationen zu diesem Vorgang

Wenn der Warteschlangenmanager erstellt und die Abfangprozesse für das Abfangen von Nachrichten und den Zugriff auf Verschlüsselungsschlüssel vorbereitet sind, können wir mit dem Befehl `setmqspl` mit dem Definieren von Zugriffsschutzrichtlinien für `QM_VERIFY_AMS` beginnen. Weitere Informationen zu diesem Befehl finden Sie in setmqspl . Jeder Richtlinienname muss mit dem Namen der Warteschlange identisch sein, auf die er angewendet werden soll.

Beispiel

Dies ist ein Beispiel für eine Richtlinie, die für die `TEST.Q` -Warteschlange definiert ist. In dem Beispiel werden Nachrichten mit dem SHA1 -Algorithmus signiert und mit dem Algorithmus AES256 verschlüsselt. `alice` ist der einzige gültige Sender, und bob ist der einzige Empfänger der Nachrichten in dieser Warteschlange:

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

Anmerkung: Die DNs stimmen exakt mit den DNs überein, die im Zertifikat des jeweiligen Benutzers aus der Schlüsseldatenbank angegeben sind.

Nächste Schritte

Geben Sie den folgenden Befehl aus, um die von Ihnen definierte Richtlinie zu überprüfen:

```
dspmqspl -m QM_VERIFY_AMS
```

Wenn Sie die Richtliniendetails als eine Gruppe von `setmqspl` -Befehlen drucken möchten, verwenden Sie die Markierung `-export` . Auf diese Weise können bereits definierte Richtlinien gespeichert werden:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Setup testen

Informationen zu diesem Vorgang

Wenn Sie verschiedene Programme unter verschiedenen Benutzern ausführen, können Sie überprüfen, ob die Anwendung ordnungsgemäß konfiguriert wurde.

Vorgehensweise

1. Wechseln Sie zum Benutzer `alice`

   Klicken Sie mit der rechten Maustaste auf `cmd.exe` und wählen Sie **Ausführen als ...** aus. Wenn Sie dazu aufgefordert werden, melden Sie sich als Benutzer `alice` an.
2. Wenn der Benutzer `alice` eine Nachricht mithilfe einer Musteranwendung einsetzte:

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. Geben Sie den Text der Nachricht ein und drücken Sie die Eingabetaste.
4. Wechseln Sie zum Benutzer bob

   Öffnen Sie ein anderes Fenster, indem Sie mit der rechten Maustaste auf `cmd.exe` klicken und **Ausführen als ...** auswählen. Wenn Sie dazu aufgefordert werden, melden Sie sich als Benutzer bob an.
5. Wenn der Benutzer bob eine Nachricht mit Hilfe einer Beispielanwendung abrufen kann:

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

Ergebnisse

Wenn die Anwendung für beide Benutzer ordnungsgemäß konfiguriert wurde, wird die Nachricht des Benutzers `alice` angezeigt, wenn bob die Anwendung "Erholen" ausführt.

8. Verschlüsselung testen

Informationen zu diesem Vorgang

Erstellen Sie eine Aliaswarteschlange, die auf die ursprüngliche Warteschlange `TEST.Q` verweist, um zu überprüfen, ob die Verschlüsselung wie erwartet ausgeführt wird. Diese Aliaswarteschlange verfügt über keine Sicherheitsrichtlinie, so dass kein Benutzer über die Informationen zum Entschlüsseln der Nachricht verfügt und daher die verschlüsselten Daten angezeigt werden.

Vorgehensweise

1. Erstellen Sie mit dem Befehl **runmqsc** für den Warteschlangenmanager QM_VERIFY_AMS eine Aliaswarteschlange.

   ```
   DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
   ```
2. Erteilen Sie bob den Zugriff zum Durchsuchen aus der Aliaswarteschlange.

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. Wenn der Benutzer `alice` eine andere Nachricht mit einer Beispielanwendung wie zuvor eingibt, wird Folgendes angezeigt:

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. Als Benutzer bob können Sie die Nachricht über die Aliaswarteschlange dieses Mal mit Hilfe einer Musteranwendung durchsuchen:

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. Wenn der Benutzer bob die Nachricht mit Hilfe einer Musteranwendung aus der lokalen Warteschlange abrufen soll, gehen Sie wie folgt vor:

```
amqsget TEST.Q QM_VERIFY_AMS
```

Ergebnisse

Die Ausgabe der Anwendung " `amqsbcg` " zeigt die verschlüsselten Daten, die sich in der Warteschlange befindet, aus der hervorgeht, dass die Nachricht verschlüsselt wurde.

Linux AIX *Leitfaden für den Schnelleinstieg für AMS unter AIX and Linux*

Verwenden Sie dieses Handbuch für die schnelle Konfiguration von Advanced Message Security, um die Nachrichtensicherheit auf AIX and Linux-Plattformen bereitzustellen. Nach Abschluss der Ausführung haben Sie eine Schlüsseldatenbank erstellt, um die Benutzeridentitäten und die definierten Richtlinien für die Signierung/Verschlüsselung für Ihren Warteschlangenmanager zu überprüfen.

Vorbereitende Schritte

Es sollten mindestens die folgenden Komponenten auf Ihrem System installiert sein:

- Laufzeit
- Server
- Beispielprogramme
- IBM Global Security Kit
- Advanced Message Security

Die Komponentennamen auf den einzelnen Plattformen finden Sie in den folgenden Abschnitten:

- Linux IBM MQ-Komponenten für Linux-Systeme
- AIX IBM MQ-Komponenten für AIX-Systeme

1. WS-Manager und eine Warteschlange erstellen

Informationen zu diesem Vorgang

In allen folgenden Beispielen wird eine Warteschlange mit dem Namen `TEST.Q` verwendet, um Nachrichten zwischen Anwendungen zu übergeben. Advanced Message Security verwendet Interceptors, um Nachrichten an dem Punkt zu signieren und zu verschlüsseln, an dem Sie in der IBM MQ-Infrastruktur über die Standardschnittstelle von IBM MQ eintreffen. Die Basiseinrichtung wird in IBM MQ vorgenommen und in den folgenden Schritten konfiguriert.

Sie können mit IBM MQ den Warteschlangenmanager QM_VERIFY_AMS und die zugehörige lokale Warteschlange mit der Bezeichnung `TEST.Q` erstellen, indem Sie alle Standardeinstellungen des Assistenten

übernehmen oder die Befehle in *MQ_INSTALLATION_PATH*/bin verwenden. Denken Sie daran, dass Sie ein Mitglied der mqm -Benutzergruppe sein müssen, um die folgenden Verwaltungsbefehle auszuführen.

Vorgehensweise

1. Einen WS-Manager erstellen

```
crtmqm QM_VERIFY_AMS
```

2. WS-Manager starten

```
strmqm QM_VERIFY_AMS
```

3. Erstellen Sie eine Warteschlange mit dem Namen TEST.Q , indem Sie den folgenden Befehl in **runmqsc** für Warteschlangenmanager QM_VERIFY_AMS eingeben.

```
DEFINE QLOCAL(TEST.Q)
```

Ergebnisse

Wenn die Prozedur erfolgreich abgeschlossen wurde, zeigt der folgende Befehl, der in **runmqsc** eingegeben wurde, Details zu TEST.Qan:

```
DISPLAY Q(TEST.Q)
```

2. Benutzer erstellen und berechtigen

Informationen zu diesem Vorgang

In diesem Beispiel werden zwei Benutzer angezeigt: alice, der Sender und bob, der Empfänger. Um die Anwendungswarteschlange verwenden zu können, müssen diese Benutzer berechtigt sein, sie zu verwenden. Außerdem müssen Sie die Zugriffsschutzrichtlinien, die wir definieren, erfolgreich verwenden, um Zugriff auf einige Systemwarteschlangen zu erhalten. Weitere Informationen zum Befehl **setmqaut** finden Sie unter **setmqaut**.

Vorgehensweise

1. Erstellen Sie die beiden Benutzer.

```
useradd alice
useradd bob
```

2. Benutzer berechtigen, eine Verbindung zum WS-Manager herzustellen und mit der Warteschlange zu arbeiten

```
setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get
```

3. Außerdem sollten Sie den beiden Benutzern die Möglichkeit geben, die Systemrichtlinienwarteschlange zu durchsuchen und Nachrichten in die Fehlerwarteschlange einzureihen.

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p alice -p bob +browse
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p alice -p bob +put
```

Achtung: IBM MQ optimiert die Leistung durch das Caching von Richtlinien, sodass Sie keine Datensätze nach Richtliniendetails auf dem SYSTEM.PROTECTION.POLICY.QUEUE in allen Fällen.

Von IBM MQ werden nicht alle verfügbaren Richtlinien zwischengespeichert. Wenn eine hohe Anzahl an Richtlinien vorhanden ist, wird von IBM MQ eine begrenzte Anzahl von Richtlinien zwischengespeichert. Wenn also der Warteschlangenmanager eine geringe Anzahl von Richtlinien definiert hat, ist es nicht erforderlich, die Option zum Durchsuchen für SYSTEM.PROTECTION.POLICY.QUEUE bereitzustellen.

Sie sollten jedoch die Berechtigung zum Durchsuchen dieser Warteschlange erteilen, falls eine hohe Anzahl an Richtlinien definiert ist, oder wenn Sie alte Clients verwenden. Die Warteschlange SYSTEM.PROTECTION.ERROR.QUEUE wird zum Einreihen von Fehlernachrichten verwendet, die vom AMS-Code generiert werden. Die Einreihungsberechtigung für diese Warteschlange wird nur überprüft, wenn Sie versuchen, eine Fehlernachricht in die Warteschlange einzureihen. Ihre Einreihungsberechtigung für die Warteschlange wird nicht überprüft, wenn Sie versuchen, eine Nachricht in eine AMS-geschützte Warteschlange einzureihen oder daraus abzurufen.

Ergebnisse

Benutzergruppen werden jetzt erstellt, und die erforderlichen Berechtigungen werden ihnen erteilt. Auf diese Weise erhalten Benutzer, die diesen Gruppen zugeordnet sind, auch die Berechtigung zum Herstellen einer Verbindung zum Warteschlangenmanager und zum Einlegen und Abrufen aus der Warteschlange.

Nächste Schritte

Um zu überprüfen, ob die Schritte ordnungsgemäß ausgeführt wurden, verwenden Sie die Beispiele `amqsput` und `amqsget` wie im Abschnitt „8. Verschlüsselung testen” auf Seite 658beschrieben.

3. Schlüsseldatenbank und Zertifikate erstellen

Informationen zu diesem Vorgang

Um die Nachricht zu verschlüsseln, benötigt der Interceptor den privaten Schlüssel des sendenden Benutzers und die öffentlichen Schlüssel des/der Empfänger (s). Daher muss die Schlüsseldatenbank der Benutzeridentitäten, die öffentlichen und privaten Schlüsseln zugeordnet sind, erstellt werden. Im realen System, in dem Benutzer und Anwendungen auf mehreren Computern verteilt sind, hat jeder Benutzer seinen eigenen privaten Schlüsselspeicher. Ebenso erstellen wir in diesem Handbuch Schlüsseldatenbanken für `alice` und bob und nutzen die Benutzerzertifikate zwischen ihnen gemeinsam.

Anmerkung: In diesem Handbuch verwenden wir Musteranwendungen, die in C geschrieben sind und die lokale Bindungen verwenden. Wenn Sie Java-Anwendungen mit Clientbindungen verwenden möchten, müssen Sie einen JKS-Keystore und Zertifikate mit dem Befehl **`keytool`** erstellen, der Teil der JRE ist (weitere Informationen finden Sie unter „Leitfaden für den Schnelleinstieg für AMS mit Java-Clients” auf Seite 669). Für alle anderen Sprachen und für Java-Anwendungen, die lokale Bindungen verwenden, sind die Schritte in diesem Handbuch korrekt.

Vorgehensweise

1. Erstellen Sie eine neue Schlüsseldatenbank für den Benutzer `alice`

```
mkdir /home/alice/.mqs -p
runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
```

Anmerkung:

- Es ist ratsam, ein sicheres Kennwort zu verwenden, um die Datenbank zu sichern.
- Der Parameter **`stash`** speichert das Kennwort in der Datei `key.sth` , die vom Interceptor zum Öffnen der Datenbank verwendet werden kann.

2. Stellen Sie sicher, dass die Schlüsseldatenbank lesbar ist

```
chmod +r /home/alice/.mqs/alicekey.kdb
```

3. Erstellen Sie ein Zertifikat, das den Benutzer `alice` für die Verwendung in der Verschlüsselung identifiziert.

```
runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd
-label Alice_Cert -dn "cn=alice,O=IBM,c=GB" -default_cert yes
```

Anmerkung:

- Im Sinne dieses Leitfadens verwenden wir selbst signiertes Zertifikat, das ohne Verwendung einer Zertifizierungsstelle erstellt werden kann. Bei Produktionssystemen ist es ratsam, keine selbst signierten Zertifikate zu verwenden, sondern sich auf Zertifikate zu stützen, die von einer Zertifizierungsstelle signiert wurden.
- Der Parameter **`label`** gibt den Namen für das Zertifikat an, in dem die Interceptors nach den erforderlichen Informationen suchen.
- Der Parameter **DN** gibt die Details zu **Definierter Name** (DN) an, die für jeden Benutzer eindeutig sein müssen.

4. Nun haben wir die Schlüsseldatenbank erstellt, wir sollten das Eigentumsrecht festlegen und sicherstellen, dass sie nicht von allen anderen Benutzern gelesen werden kann.

```
chown alice /home/alice/.mqs/alicekey.kdb /home/alice/.mqs/alicekey.sth
chmod 600 /home/alice/.mqs/alicekey.kdb /home/alice/.mqs/alicekey.sth
```

5. Wiederholen Sie die Schritte 1 bis 4 für Benutzer bob

Ergebnisse

Die beiden Benutzer `alice` und bob verfügen nun jeweils über ein selbst signiertes Zertifikat.

4. Keystore.conf erstellen

Informationen zu diesem Vorgang

Advanced Message Security-Interceptors müssen auf das Verzeichnis verweisen, in dem sich die Schlüsseldatenbanken und Zertifikate befinden. Dies erfolgt über die Datei `keystore.conf`, die diese Informationen im Klartextformat enthält. Jeder Benutzer muss über eine separate `keystore.conf`-Datei im Ordner `.mqs` verfügen. Dieser Schritt muss für `alice` und bobausgeführt werden.

Der Inhalt von `keystore.conf` muss das folgende Format haben:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

Beispiel

Für dieses Szenario wird der Inhalt des `keystore.conf` wie folgt aussehen:

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

Anmerkung:

- Der Pfad zu der Keystore-Datei muss ohne Dateierweiterung angegeben werden.
- Es gibt die folgenden Keystore-Formate: CMS (Cryptographic Message Syntax), JKS (Java Keystore) und JCEKS (Java Cryptographic Extension Keystore). Weitere Informationen hierzu finden Sie unter „Struktur der Keystore-Konfigurationsdatei (keystore.conf) für AMS” auf Seite 684.
- `HOME/.mqs/keystore.conf` ist die Standardposition, in der Advanced Message Security nach der `keystore.conf`-Datei sucht. Informationen zur Verwendung einer nicht standardmäßigen Position für die `keystore.conf` finden Sie unter „Keystores und Zertifikate mit AMS verwenden” auf Seite 682.

5. Zertifikate gemeinsam nutzen

Informationen zu diesem Vorgang

Geben Sie die Zertifikate zwischen den beiden Schlüsseldatenbanken frei, so dass jeder Benutzer die andere identifizieren kann. Dazu wird jedes öffentliche Zertifikat jedes Benutzers in eine Datei extrahiert, die dann zur Schlüsseldatenbank des anderen Benutzers hinzugefügt wird.

Anmerkung: Gehen Sie vorsichtig vor, um die Option *extract* zu verwenden, und nicht die Option *export* . *Extrahieren* ruft den öffentlichen Schlüssel des Benutzers ab, während *export* sowohl den öffentlichen als auch den privaten Schlüssel erhält. Wenn Sie *export* versehentlich verwenden, würde Ihre Anwendung vollständig durch die Weitergabe des privaten Schlüssels beeinträchtigt.

Vorgehensweise

1. Extrahieren Sie das Zertifikat, das `alice` identifiziert, in eine externe Datei:

```
runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
-target alice_public.arm
```

2. Fügen Sie das Zertifikat dem `bob's` -Keystore hinzu:

```
runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file ali
ce_public.arm
```

3. Wiederholen Sie den Schritt für `bob`:

```
runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
bob_public.arm
```

4. Fügen Sie das Zertifikat für `bob` zum `alice's` -Keystore hinzu:

```
runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
bob_public.arm
```

Ergebnisse

Die beiden Benutzer `alice` und `bob` sind jetzt in der Lage, einander erfolgreich zu identifizieren, wenn sie selbst signierte und gemeinsam signierte Zertifikate erstellt haben.

Nächste Schritte

Stellen Sie sicher, dass sich ein Zertifikat im Keystore befindet, indem Sie die folgenden Befehle ausführen, die die zugehörigen Details ausgeben:

```
runmqakm -cert -details -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert
runmqakm -cert -details -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert
```

6. Warteschlangenrichtlinie definieren

Informationen zu diesem Vorgang

Wenn der Warteschlangenmanager erstellt und die Abfangprozesse für das Abfangen von Nachrichten und den Zugriff auf Verschlüsselungsschlüssel vorbereitet sind, können wir mit dem Befehl `setmqspl` mit dem Definieren von Zugriffsschutzrichtlinien für `QM_VERIFY_AMS` beginnen. Weitere Informationen zu diesem Befehl finden Sie in setmqspl . Jeder Richtlinienname muss mit dem Namen der Warteschlange identisch sein, auf die er angewendet werden soll.

Beispiel

Dies ist ein Beispiel für eine Richtlinie, die für die TEST.Q -Warteschlange definiert ist. In diesem Beispiel werden Nachrichten vom Benutzer alice mit dem SHA1 -Algorithmus signiert und mit dem 256-Bit-Algorithmus von AES verschlüsselt. alice ist der einzige gültige Sender, und bob ist der einzige Empfänger der Nachrichten in dieser Warteschlange:

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

Anmerkung: Die DNs stimmen exakt mit den DNs überein, die im Zertifikat des jeweiligen Benutzers aus der Schlüsseldatenbank angegeben sind.

Nächste Schritte

Geben Sie den folgenden Befehl aus, um die von Ihnen definierte Richtlinie zu überprüfen:

```
dspmqspl -m QM_VERIFY_AMS
```

Wenn Sie die Richtliniendetails als eine Gruppe von setmqspl -Befehlen drucken möchten, verwenden Sie die Markierung -export . Auf diese Weise können bereits definierte Richtlinien gespeichert werden:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Setup testen

Informationen zu diesem Vorgang

Wenn Sie verschiedene Programme unter verschiedenen Benutzern ausführen, können Sie überprüfen, ob die Anwendung ordnungsgemäß konfiguriert wurde.

Vorgehensweise

1. Wechseln Sie in das Verzeichnis, das die Beispiele enthält. Wenn MQ in einer anderen Position als der Standardposition installiert ist, kann dies an einem anderen Ort liegen.

   ```
   cd /opt/mqm/samp/bin
   ```

2. Wechseln Sie zum Benutzer alice

   ```
   su alice
   ```

3. Geben Sie als Benutzer alice eine Nachricht mit einer Beispielanwendung ein:

   ```
   ./amqsput TEST.Q QM_VERIFY_AMS
   ```

4. Geben Sie den Text der Nachricht ein und drücken Sie die Eingabetaste.
5. Stoppen Sie die Ausführung als Benutzer alice

   ```
   exit
   ```

6. Wechseln Sie zum Benutzer bob

   ```
   su bob
   ```

7. Geben Sie als Benutzer bob eine Nachricht mit Hilfe einer Beispielanwendung an:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Ergebnisse

Wenn die Anwendung für beide Benutzer ordnungsgemäß konfiguriert wurde, wird die Nachricht des Benutzers `alice` angezeigt, wenn bob die Anwendung "Erholen" ausführt.

8. Verschlüsselung testen

Informationen zu diesem Vorgang

Erstellen Sie eine Aliaswarteschlange, die auf die ursprüngliche Warteschlange TEST.Q verweist, um zu überprüfen, ob die Verschlüsselung wie erwartet ausgeführt wird. Diese Aliaswarteschlange verfügt über keine Sicherheitsrichtlinie, so dass kein Benutzer über die Informationen zum Entschlüsseln der Nachricht verfügt und daher die verschlüsselten Daten angezeigt werden.

Vorgehensweise

1. Erstellen Sie mit dem Befehl **runmqsc** für den Warteschlangenmanager QM_VERIFY_AMS eine Aliaswarteschlange.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Erteilen Sie bob den Zugriff zum Durchsuchen aus der Aliaswarteschlange.

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. Wenn der Benutzer `alice` eine andere Nachricht mit einer Beispielanwendung wie zuvor eingibt, wird Folgendes angezeigt:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. Als Benutzer bob können Sie die Nachricht über die Aliaswarteschlange dieses Mal mit Hilfe einer Musteranwendung durchsuchen:

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. Wenn der Benutzer bob die Nachricht mit Hilfe einer Musteranwendung aus der lokalen Warteschlange abrufen soll, gehen Sie wie folgt vor:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Ergebnisse

In der Ausgabe der Anwendung " `amqsbcg` " werden die verschlüsselten Daten angezeigt, die sich in der Warteschlange für die Verschlüsselung der Nachricht befindet.

z/OS *Beispiel für AMS-Konfigurationen unter z/OS*

Dieser Abschnitt enthält Beispielkonfigurationen für Richtlinien und Zertifikate für Advanced Message Security-Warteschlangenszenarios unter z/OS.

Details zur Konfiguration von Advanced Message Securityfinden Sie unter Advanced Message Security for z/OSkonfigurieren .

Die Beispiele beziehen sich auf die erforderlichen Advanced Message Security-Richtlinien und die digitalen Zertifikate, die relativ zu Benutzern und Schlüsselringen vorhanden sein müssen. In den Beispielen wird davon ausgegangen, dass die Benutzer, die an den Szenarios beteiligt sind, anhand der Anweisungen unter Benutzern Ressourcenberechtigungen für Advanced Message Security erteilen konfiguriert wurden.

V 9.2.0 Weitere Informationen finden Sie unter Server-zu-Server-Nachrichtenkanalabfangbeispiele ab IBM MQ 9.1.3.

z/OS *Steuerung ferner Warteschlangen für integritätsgeschützte Nachrichten für AMS unter z/OS*
In diesem Beispiel finden Sie Informationen zu Richtlinien und Zertifikaten für Advanced Message Security, die erforderlich sind, um Nachrichten mit dem Datenschutzniveau 'Integrity' an eine Warteschlange zu senden und von dort abzurufen, bei der es sich um eine lokale Warteschlangen für das Einreihen und Abrufen durch Anwendungen handelt.

Der Beispielwarteschlangenmanager und die Warteschlange sind:

```
BNK6        - Queue manager
FIN.XFER.Q7 - Local queue
```

Diese Benutzer werden verwendet:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

Erstellen Sie die Benutzerzertifikate.

In diesem Beispiel ist nur ein Benutzerzertifikat erforderlich. Dies ist das Zertifikat des sendenden Benutzers, das für die Unterzeichnung integritätsgeschützter Nachrichten erforderlich ist. Der sendende Benutzer ist 'TELLER5'.

Das Zertifikat der Zertifizierungsinstanz (CA) ist ebenfalls erforderlich. Das CA-Zertifikat ist das Zertifikat der Berechtigung, die das Zertifikat des Benutzers ausgestellt hat. Dies kann eine Kette von Zertifikaten sein. Ist dies der Fall, sind alle Zertifikate in der Kette im Schlüsselring des Advanced Message Security-Taskbenutzers (hier WMQBNK6) erforderlich.

Ein CA-Zertifikat kann mit dem RACF-Befehl RACDCERT erstellt werden. Dieses Zertifikat wird zum Ausgeben von Benutzerzertifikaten verwendet. Beispiel:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Mit diesem RACDCERT-Befehl wird ein CA-Zertifikat erstellt, mit dem dann ein Benutzerzertifikat für Benutzer 'TELLER5' ausgegeben werden kann. Beispiel:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Ihre Installation hat Prozeduren zum Auswählen oder Erstellen eines CA-Zertifikats sowie Prozeduren zum Ausstellen von Zertifikaten und zum Verteilen dieser Zertifikate auf relevante Systeme.

Beim Exportieren und Importieren dieser Zertifikate sind für Advanced Message Security die folgenden Zertifikate erforderlich:

- Das CA-Zertifikat (Kette).
- Das Benutzerzertifikat und sein privater Schlüssel.

Wenn Sie RACF verwenden, können Zertifikate mit dem Befehl RACDCERT EXPORT in ein Dataset exportiert und mit dem Befehl RACDCERT ADD aus dem Dataset importiert werden. Weitere Informationen zu diesen und anderen RACDCERT-Befehlen finden Sie im Handbuch *z/OS: Security Server RACF Command Language Reference*.

Die Zertifikate sind in diesem Fall auf dem z/OS-System erforderlich, auf dem Warteschlangenmanager BNK6 ausgeführt wird.

Wenn die Zertifikate in das z/OS-System importiert wurden, auf dem BNK6 ausgeführt wird, ist für das Benutzerzertifikat das Attribut TRUST erforderlich. Der Befehl RACDCERT ALTER kann verwendet werden, um das Attribut TRUST dem Zertifikat hinzuzufügen. Beispiel:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

In diesem Beispiel ist kein Zertifikat für den Empfängerbenutzer erforderlich.

Zertifikate mit relevanten Schlüsselringen verbinden

Wenn die erforderlichen Zertifikate erstellt oder importiert und als vertrauenswürdig festgelegt wurden, müssen sie mit den entsprechenden Benutzerschlüsselringen auf dem z/OS-System verbunden werden, auf dem BNK6 ausgeführt wird. Verwenden Sie zum Erstellen der Schlüsselringe die RACDCERT ADD-RING-Befehle:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Dadurch wird ein Schlüsselring für den Advanced Message Security-Taskbenutzer WMQBNK6 und ein Schlüsselring für den sendenden Benutzer 'TELLER5' erstellt. Beachten Sie, dass der Schlüsselringname drq.ams.keyring obligatorisch ist und der Name die Groß-/Kleinschreibung beachtet.

Wenn die Schlüsselringe erstellt wurden, können die entsprechenden Zertifikate verbunden werden:

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

Das sendende Benutzerzertifikat muss als DEFAULT verbunden sein. Wenn der sendende Benutzer mehr als ein Zertifikat in seinem drq.ams.keyring hat, wird das Standardzertifikat zu Signierungszwecken verwendet.

Die Erstellung und Änderung von Zertifikaten wird von Advanced Message Security erst erkannt, wenn der Queue Manager gestoppt und erneut gestartet wird oder wenn der z/OS **MODIFY**-Befehl zum Aktualisieren der Advanced Message Security-Zertifikatskonfiguration verwendet wird. Beispiel:

```
F BNK6AMSM,REFRESH KEYRING
```

Advanced Message Security-Richtlinie erstellen

In diesem Beispiel werden die Nachrichten mit dem Datenschutzniveau 'Integrity' von einer Anwendung, die als Benutzer 'TELLER5' ausgeführt wird, in die Warteschlange FIN.XFER.Q7 eingereiht und von einer Anwendung, die als Benutzer 'FINADM2' ausgeführt wird, aus der gleichen Warteschlange abgerufen, sodass nur eine Advanced Message Security-Richtlinie erforderlich ist.

Advanced Message Security-Richtlinien werden mit dem Dienstprogramm CSQ0UTIL erstellt, wie unter Dienstprogramm für Nachrichtensicherheitsrichtlinie (CSQ0UTIL) dokumentiert ist.

Verwenden Sie das Dienstprogramm CSQ0UTIL, um den folgenden Befehl auszuführen:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In dieser Richtlinie wird der WS-Manager als BNK6 bezeichnet. Der Richtlinienname und die zugehörige Warteschlange sind FIN.XFER.Q7. Der Algorithmus, der für die Generierung der Signatur des Absenders

verwendet wird, ist MD5, und der definierte Name (DN) des sendenden Benutzers lautet 'CN=Teller5, O=BCO, C=US'.

Nach dem Definieren der Richtlinie starten Sie den Warteschlangenmanager BNK6 erneut oder verwenden den z/OS-Befehl **MODIFY**, um die Konfiguration der Advanced Message Security-Richtlinie zu aktualisieren. Beispiel:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *Steuerung ferner Warteschlangen für Nachrichten mit Datenschutz für AMS unter z/OS*

In diesem Beispiel finden Sie Informationen zu Richtlinien und Zertifikaten für Advanced Message Security, die erforderlich sind, um Nachrichten mit dem Datenschutzniveau 'Privacy' an eine Warteschlange zu senden und von dort abzurufen, bei der es sich um eine lokale Warteschlangen für das Einreihen und Abrufen durch Anwendungen handelt. Datenschutz-geschützte Nachrichten werden signiert und verschlüsselt.

Der Beispielwarteschlangenmanager und die lokale Warteschlange lauten wie folgt:

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

Diese Benutzer werden verwendet:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

Gehen Sie wie folgt vor, um dieses Szenario zu konfigurieren:

Erstellen Sie die Benutzerzertifikate.

In diesem Beispiel sind zwei Benutzerzertifikate erforderlich. Dies sind das Zertifikat des sendenden Benutzers, das zum Signieren von Nachrichten benötigt wird, und das Zertifikat des Empfängerbenutzers, das zum Verschlüsseln und Entschlüsseln der Nachrichtendaten benötigt wird. Der sendende Benutzer ist 'TELLER5', und der Empfängerbenutzer ist 'FINADM2'.

Das Zertifikat der Zertifizierungsinstanz (CA) ist ebenfalls erforderlich.Das CA-Zertifikat ist das Zertifikat der Berechtigung, die das Zertifikat des Benutzers ausgestellt hat. Dies kann eine Kette von Zertifikaten sein. Ist dies der Fall, sind alle Zertifikate in der Kette im Schlüsselring des Advanced Message Security-Taskbenutzers (hier WMQBNK6) erforderlich.

Ein CA-Zertifikat kann mit dem RACF-Befehl RACDCERT erstellt werden. Dieses Zertifikat wird zum Ausgeben von Benutzerzertifikaten verwendet. Beispiel:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Mit diesem Befehl RACDCERT wird ein CA-Zertifikat erstellt, das dann zur Ausgabe von Benutzerzertifikaten für die Benutzer 'TELLER5' und 'FINADM2' verwendet werden kann. Beispiel:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Ihre Installation hat Prozeduren zum Auswählen oder Erstellen eines CA-Zertifikats sowie Prozeduren zum Ausstellen von Zertifikaten und zum Verteilen dieser Zertifikate auf relevante Systeme.

Beim Exportieren und Importieren dieser Zertifikate sind für Advanced Message Security die folgenden Zertifikate erforderlich:

- Das CA-Zertifikat (Kette).
- Das sendende Benutzerzertifikat und sein privater Schlüssel.
- Das Empfängerbenutzerzertifikat und sein privater Schlüssel.

Wenn Sie RACF verwenden, können Zertifikate mit dem Befehl RACDCERT EXPORT in ein Dataset exportiert und mit dem Befehl RACDCERT ADD aus dem Dataset importiert werden. Weitere Informationen zu diesen und anderen RACDCERT-Befehlen finden Sie unter RACDCERT (Manage RACF digital certificates) in der Veröffentlichung *z/OS: Security Server RACF Command Language Reference*.

Die Zertifikate in diesem Fall sind auf dem z/OS-System erforderlich, auf dem Warteschlangenmanager BNK6 ausgeführt wird.

Wenn die Zertifikate in das z/OS-System importiert wurden, auf dem BNK6 ausgeführt wird, ist für die Benutzerzertifikate das Attribut TRUST erforderlich. Der Befehl RACDCERT ALTER kann verwendet werden, um das Attribut TRUST dem Zertifikat hinzuzufügen. Beispiel:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Zertifikate mit relevanten Schlüsselringen verbinden

Wenn die erforderlichen Zertifikate erstellt oder importiert und als vertrauenswürdig festgelegt wurden, müssen sie mit den entsprechenden Benutzerschlüsselringen auf dem z/OS-System verbunden werden, auf dem BNK6 ausgeführt wird. Verwenden Sie den Befehl RACDCERT ADDRING, um die Schlüsselringe zu erstellen:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Dadurch wird ein Schlüsselring für den Benutzer der Advanced Message Security-Task erstellt und es werden Schlüsselringe für die sendenden und empfangenden Benutzer erstellt. Beachten Sie, dass der Schlüsselringname drq.ams.keyring obligatorisch ist und der Name die Groß-/Kleinschreibung beachtet.

Wenn die Schlüsselringe erstellt wurden, können die entsprechenden Zertifikate verbunden werden.

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))

RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

Die sendenden und die Empfänger-Benutzerzertifikate müssen als DEFAULT verbunden sein. Wenn der Benutzer über mehr als ein Zertifikat in seinem drq.ams.keyring verfügt, wird das Standardzertifikat zum Signieren und Entschlüsseln verwendet.

Das Zertifikat des Empfängers muss mit USAGE(SITE) auch mit dem Schlüsselring für den Benutzer der Advanced Message Security-Task verbunden werden. Dies liegt daran, dass die Task "Advanced Message Security" beim Verschlüsseln der Nachrichtendaten den öffentlichen Schlüssel des Empfängers benötigt. Die Klausel USAGE (SITE) verhindert, dass der private Schlüssel im Schlüsselring zugänglich ist.

Die Erstellung und Änderung von Zertifikaten wird von Advanced Message Security erst erkannt, wenn der Queue Manager gestoppt und erneut gestartet wird oder wenn der z/OS **MODIFY**-Befehl zum Aktualisieren der Advanced Message Security-Zertifikatskonfiguration verwendet wird. Beispiel:

```
F BNK6AMSM,REFRESH KEYRING
```

Advanced Message Security-Richtlinie erstellen

In diesem Beispiel werden die Nachrichten mit dem Datenschutzniveau 'Privacy' von einer Anwendung, die als Benutzer 'TELLER5' ausgeführt wird, in die Warteschlange FIN.XFER.Q8 eingereiht und von einer Anwendung, die als Benutzer 'FINADM2' ausgeführt wird, aus der gleichen Warteschlange abgerufen, sodass nur eine Advanced Message Security-Richtlinie erforderlich ist.

Advanced Message Security-Richtlinien werden mit dem Dienstprogramm CSQ0UTIL erstellt, wie unter Dienstprogramm für Nachrichtensicherheitsrichtlinie (CSQ0UTIL) dokumentiert ist.

Verwenden Sie das Dienstprogramm CSQ0UTIL, um den folgenden Befehl auszuführen:

```
setmqspl -m BNK6 -p FIN.XFER.Q8 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r CN=Fi
nAdm2,O=BCO,C=US
```

In dieser Richtlinie wird der WS-Manager als BNK6 bezeichnet. Der Richtlinienname und die zugehörige Warteschlange sind FIN.XFER.Q8. Der Algorithmus, der zum Generieren der Signatur des Absenders verwendet wird, lautet SHA1, und der definierte Name (DN) des sendenden Benutzers lautet 'CN=Teller5, O=BCO, C=US', und der Empfängerbenutzer ist 'CN=FinAdm2, O=BCO, C=US'. Der Algorithmus, der zum Verschlüsseln der Nachrichtendaten verwendet wird, ist 3DES.

Nach dem Definieren der Richtlinie starten Sie den Warteschlangenmanager BNK6 erneut oder verwenden den z/OS-Befehl **MODIFY**, um die Konfiguration der Advanced Message Security-Richtlinie zu aktualisieren. Beispiel:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *Steuerung ferner Warteschlangen für integritätsgeschützte Nachrichten für AMS unter z/OS*

In diesem Beispiel finden Sie Informationen zu Richtlinien und Zertifikaten für Advanced Message Security, die erforderlich sind, um Nachrichten mit den Datenschutzniveau 'Integrity' an Warteschlangen zu senden und aus diesen abzurufen, die von zwei verschiedenen Warteschlangenmanagern verwaltet werden. Die beiden Warteschlangenmanager können auf dem gleichen z/OS-System oder auf unterschiedlichen z/OS-Systemen ausgeführt werden oder ein Warteschlangenmanager kann sich auf einem verteilten System befinden, auf dem Advanced Message Security ausgeführt wird.

Die Beispielwarteschlangenmanager und -warteschlangen lauten wie folgt:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Hinweis: In diesem Beispiel werden die beiden Warteschlangenmanager BNK6 und BNK7 auf verschiedenen z/OS-Systemen ausgeführt.

Diese Benutzer werden verwendet:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMStask user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

Gehen Sie wie folgt vor, um dieses Szenario zu konfigurieren:

Erstellen Sie die Benutzerzertifikate.

In diesem Beispiel ist nur ein Benutzerzertifikat erforderlich. Dies ist das Zertifikat des sendenden Benutzers, das zum Signieren der Integritätsbedingungsnachricht erforderlich ist. Der sendende Benutzer ist 'TELLER5'.

Das Zertifikat der Zertifizierungsinstanz (CA) ist ebenfalls erforderlich.Das CA-Zertifikat ist das Zertifikat der Berechtigung, die das Zertifikat des Benutzers ausgestellt hat. Dies kann eine Kette von Zertifikaten sein. Ist dies der Fall, sind alle Zertifikate im Schlüsselring des Advanced Message Security-Taskbenutzers erforderlich, hier also Benutzer WMQBNK7.

Ein CA-Zertifikat kann mit dem RACF-Befehl RACDCERT erstellt werden. Dieses Zertifikat wird zum Ausgeben von Benutzerzertifikaten verwendet. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Mit diesem Befehl RACDCERT wird ein CA-Zertifikat erstellt, das dann zur Ausgabe des Benutzerzertifikats für den Benutzer 'TELLER5' verwendet werden kann. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Ihre Installation hat Prozeduren zum Auswählen oder Erstellen eines CA-Zertifikats sowie Prozeduren zum Ausstellen von Zertifikaten und zum Verteilen dieser Zertifikate auf relevante Systeme.

Wenn Sie diese Zertifikate exportieren und importieren sind für Advanced Message Security die folgenden Zertifikate erforderlich:

- Das CA-Zertifikat (Kette).
- Das sendende Benutzerzertifikat und sein privater Schlüssel.

Wenn Sie RACF verwenden, können Zertifikate mit dem Befehl RACDCERT EXPORT in ein Dataset exportiert und mit dem Befehl RACDCERT ADD aus dem Dataset importiert werden. Weitere Informationen zu diesen und anderen RACDCERT-Befehlen finden Sie unter RACDCERT (Manage RACF digital certificates) in der Veröffentlichung *z/OS: Security Server RACF Command Language Reference*.

Die Zertifikate in diesem Fall sind auf dem z/OS-System erforderlich, auf dem Warteschlangenmanager BNK6 und BNK7 ausgeführt werden.

In diesem Beispiel muss das Sendezertifikat in das z/OS-System importiert werden, auf dem BNK6 ausgeführt wird, und das CA-Zertifikat muss in das z/OS-System importiert werden, auf dem BNK7 ausgeführt wird. Wenn die Zertifikate importiert wurden, erfordert das Benutzerzertifikat das Attribut TRUST. Der Befehl RACDCERT ALTER kann verwendet werden, um das Attribut TRUST dem Zertifikat hinzuzufügen. Beispiel für BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

Zertifikate mit relevanten Schlüsselringen verbinden

Wenn die erforderlichen Zertifikate erstellt oder importiert und als vertrauenswürdig festgelegt wurden, müssen Sie mit den zugehörigen Benutzerschlüsselringen auf dem z/OS-System verbunden werden, auf dem BNK6 und BNK7 ausgeführt wird.

Um die Schlüsselringe zu erstellen, verwenden Sie den Befehl RACDCERT ADDRING auf BNK6:

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Dadurch wird ein Schlüsselring für den sendenden Benutzer auf BNK6 erstellt. Beachten Sie, dass der Schlüsselringname drq.ams.keyring obligatorisch ist und der Name die Groß-/Kleinschreibung beachtet.

Auf BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

Dadurch wird ein Schlüsselring für den Benutzer der Advanced Message Security-Task unter BNK7 erstellt. Für 'TELLER5' auf BNK7 ist kein Benutzerschlüsselring erforderlich.

Wenn die Schlüsselringe erstellt wurden, können die entsprechenden Zertifikate verbunden werden.

Auf BNK6:

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

Auf BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

Das sendende Benutzerzertifikat muss als DEFAULT verbunden sein. Wenn der sendende Benutzer mehr als ein Zertifikat in seinem drq.ams.keyring hat, wird das Standardzertifikat zu Signierungszwecken verwendet.

Die Erstellung und Änderung von Zertifikaten wird von Advanced Message Security erst erkannt, wenn der Queue Manager gestoppt und erneut gestartet wird oder wenn der z/OS **MODIFY**-Befehl zum Aktualisieren der Advanced Message Security-Zertifikatskonfiguration verwendet wird. For example:

Auf BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

Auf BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Advanced Message Security-Richtlinien erstellen

In diesem Beispiel werden Nachrichten mit dem Datenschutzniveau 'Integrity' von einer Anwendung, die als Benutzer 'TELLER5' ausgeführt wird, in die ferne Warteschlange FIN.XFER.Q7 auf BNK6 gestellt und durch eine Anwendung, die als Benutzer 'FINADM2' ausgeführt wird, aus der lokalen Warteschlange FIN.RCPT.Q7 auf BNK7 abgerufen, sodass zwei Advanced Message Security-Richtlinien erforderlich sind.

Advanced Message Security-Richtlinien werden mit dem Dienstprogramm CSQ0UTIL erstellt, wie unter Dienstprogramm für Nachrichtensicherheitsrichtlinie (CSQ0UTIL) dokumentiert ist.

Verwenden Sie das Dienstprogramm CSQ0UTIL, um den folgenden Befehl auszuführen, um eine Integritätsrichtlinie für die ferne Warteschlange auf BNK6 zu definieren:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In dieser Richtlinie wird der WS-Manager als BNK6 bezeichnet. Der Richtlinienname und die zugehörige Warteschlange sind FIN.XFER.Q7. Der Algorithmus, der für die Generierung der Signatur des Absenders verwendet wird, ist MD5, und der definierte Name (DN) des sendenden Benutzers lautet 'CN=Teller5, O=BCO, C=US'.

Verwenden Sie außerdem das Dienstprogramm CSQ0UTIL, um den folgenden Befehl auszuführen, um eine Integritätsrichtlinie für die lokale Warteschlange auf BNK7 zu definieren:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In dieser Richtlinie wird der WS-Manager als BNK7 identifiziert. Der Richtlinienname und die zugehörige Warteschlange sind FIN.RCPT.Q7. Der für die Signatur des Absenders erwartete Algorithmus ist MD5, und der definierte Name (DN) des sendenden Benutzers soll 'CN=Teller5, O=BCO, C=US' sein.

Nach dem Definieren der beiden Richtlinien starten Sie die Warteschlangenmanager BNK6 und BNK7 erneut oder verwenden den z/OS-Befehl **MODIFY**, um die Konfigurationen der Advanced Message Security-Richtlinie zu aktualisieren. For example:

Auf BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

Auf BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

z/OS *Steuerung ferner Warteschlangen für Nachrichten mit Datenschutz für AMS unter z/OS*
In diesem Beispiel finden Sie Informationen zu Richtlinien und Zertifikaten für Advanced Message Security, die erforderlich sind, um Nachrichten mit den Datenschutzniveau 'Privacy' an Warteschlangen zu senden und aus diesen abzurufen, die von zwei verschiedenen Warteschlangenmanagern verwaltet werden. Die beiden Warteschlangenmanager können auf dem gleichen z/OS-System oder auf unterschiedlichen z/OS-Systemen ausgeführt werden oder ein Warteschlangenmanager kann sich auf einem verteilten System befinden, auf dem Advanced Message Security ausgeführt wird.

Die Beispielwarteschlangenmanager und -warteschlangen lauten wie folgt:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Hinweis: Für dieses Beispiel sind BNK6 und BNK7 Warteschlangenmanager, die auf verschiedenen z/OS-Systemen mit dem gleichen Namen ausgeführt werden.

Diese Benutzer werden verwendet:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

Gehen Sie wie folgt vor, um dieses Szenario zu konfigurieren:

Erstellen Sie die Benutzerzertifikate.

In diesem Beispiel sind zwei Benutzerzertifikate erforderlich. Dies sind das Zertifikat des sendenden Benutzers, das zum Signieren von Nachrichten benötigt wird, und das Zertifikat des Empfängerbenutzers, das zum Verschlüsseln und Entschlüsseln der Nachrichtendaten benötigt wird. Der sendende Benutzer ist 'TELLER5', und der Empfängerbenutzer ist 'FINADM2'.

Das Zertifikat der Zertifizierungsinstanz (CA) ist ebenfalls erforderlich.Das CA-Zertifikat ist das Zertifikat der Berechtigung, die das Zertifikat des Benutzers ausgestellt hat. Dies kann eine Kette von Zertifikaten sein. Ist dies der Fall, sind alle Zertifikate im Schlüsselring des Advanced Message Security-Taskbenutzers erforderlich, hier also Benutzer WMQBNK7.

Ein CA-Zertifikat kann mit dem RACF-Befehl RACDCERT erstellt werden. Dieses Zertifikat wird zum Ausgeben von Benutzerzertifikaten verwendet. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Mit diesem Befehl RACDCERT wird ein CA-Zertifikat erstellt, das dann zur Ausgabe von Benutzerzertifikaten für die Benutzer 'TELLER5' und 'FINADM2' verwendet werden kann. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Ihre Installation hat Prozeduren zum Auswählen oder Erstellen eines CA-Zertifikats sowie Prozeduren zum Ausstellen von Zertifikaten und zum Verteilen dieser Zertifikate auf relevante Systeme.

Beim Exportieren und Importieren dieser Zertifikate sind für Advanced Message Security die folgenden Zertifikate erforderlich:

- Das CA-Zertifikat (Kette).
- Das sendende Benutzerzertifikat und sein privater Schlüssel.
- Das Empfängerbenutzerzertifikat und sein privater Schlüssel.

Wenn Sie RACF verwenden, können Zertifikate mit dem Befehl RACDCERT EXPORT in ein Dataset exportiert und mit dem Befehl RACDCERT ADD aus dem Dataset importiert werden.

Weitere Informationen zu diesen und anderen RACDCERT-Befehlen finden Sie unter RACDCERT (Manage RACF digital certificates) in der Veröffentlichung *z/OS: Security Server RACF Command Language Reference*.

Die Zertifikate in diesem Fall sind auf dem z/OS-System erforderlich, auf dem Warteschlangenmanager BNK6 und BNK7 ausgeführt werden.

In diesem Beispiel müssen die Sende- und Empfängerzertifikate in das z/OS-System importiert werden, auf dem BNK6 ausgeführt wird, und die CA- und Empfängerzertifikate müssen in das z/OS-System importiert werden, auf dem BNK7 ausgeführt wird. Wenn die Zertifikate importiert wurden, ist für die Benutzerzertifikate das Attribut TRUST erforderlich. Der Befehl RACDCERT ALTER kann verwendet werden, um das Attribut TRUST dem Zertifikat hinzuzufügen. For example:

Auf BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Auf BNK7:

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Zertifikate mit relevanten Schlüsselringen verbinden

Wenn die erforderlichen Zertifikate erstellt oder importiert und als vertrauenswürdig festgelegt wurden, müssen Sie mit den zugehörigen Benutzerschlüsselringen auf den z/OS-Systemen verbunden werden, auf denen BNK6 und BNK7 ausgeführt wird.

Verwenden Sie den Befehl RACDCERT ADDRING, um die Schlüsselringe zu erstellen:

Auf BNK6:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Dadurch wird ein Schlüsselring für den Advanced Message Security-Taskbenutzer und ein Schlüsselring für den sendenden Benutzer auf BNK6 erstellt. Beachten Sie, dass der Schlüsselringname drq.ams.keyring obligatorisch ist und der Name die Groß-/Kleinschreibung beachtet.

Auf BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Dadurch wird ein Schlüsselring für den Advanced Message Security-Taskbenutzer und ein Schlüsselring für den empfangenden Benutzer auf BNK7 erstellt.

Wenn die Schlüsselringe erstellt wurden, können die entsprechenden Zertifikate verbunden werden.

Auf BNK6:

```
RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))

RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

Auf BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))

RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

Die sendenden und die Empfänger-Benutzerzertifikate müssen als DEFAULT verbunden sein. Wenn ein Benutzer mehr als ein Zertifikat in seinem drq.ams.keyring hat, wird das Standardzertifikat für die Signierung und Verschlüsselung/Entschlüsselung verwendet.

Auf BNK6 muss das Zertifikat des Empfängers auch mit dem Schlüsselring Advanced Message Security-Taskbenutzers mit der Klausel USAGE(SITE) verbunden sein. Dies liegt daran, dass die Task "Advanced Message Security" beim Verschlüsseln der Nachrichtendaten den öffentlichen Schlüssel des Empfängers benötigt. Die Klausel USAGE (SITE) verhindert, dass der private Schlüssel im Schlüsselring zugänglich ist.

Die Erstellung und Änderung von Zertifikaten wird von Advanced Message Security erst erkannt, wenn der Queue Manager gestoppt und erneut gestartet wird oder wenn der z/OS **MODIFY**-Befehl zum Aktualisieren der Advanced Message Security-Zertifikatskonfiguration verwendet wird. For example:

Auf BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

Auf BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Advanced Message Security-Richtlinien erstellen

In diesem Beispiel werden Nachrichten mit dem Datenschutzniveau 'Privacy' von einer Anwendung, die als Benutzer 'TELLER5' ausgeführt wird, in die ferne Warteschlange FIN.XFER.Q7 auf BNK6 gestellt und

durch eine Anwendung, die als Benutzer 'FINADM2' ausgeführt wird, aus der lokalen Warteschlange FIN.RCPT.Q7 auf BNK7 abgerufen, sodass zwei Advanced Message Security-Richtlinien erforderlich sind.

Advanced Message Security-Richtlinien werden mit dem Dienstprogramm CSQ0UTIL erstellt, wie unter Dienstprogramm für Nachrichtensicherheitsrichtlinie (CSQ0UTIL) dokumentiert ist.

Verwenden Sie das Dienstprogramm CSQ0UTIL, um den folgenden Befehl auszuführen, um eine Datenschutzrichtlinie für die ferne Warteschlange auf BNK6 zu definieren:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r CN=Fi
nAdm2,O=BCO,C=US
```

In dieser Richtlinie wird der WS-Manager als BNK6 bezeichnet. Der Richtlinienname und die zugehörige Warteschlange sind FIN.XFER.Q7. Der Algorithmus, der zum Generieren der Signatur des Absenders verwendet wird, lautet SHA1, der definierte Name (DN) des sendenden Benutzers lautet 'CN=Teller5, O=BCO, C=US', und der Empfängerbenutzer ist 'CN=FinAdm2, O=BCO, C=US'. Der Algorithmus, der zum Verschlüsseln der Nachrichtendaten verwendet wird, ist 3DES.

Verwenden Sie außerdem das Dienstprogramm CSQ0UTIL, um den folgenden Befehl auszuführen, um eine Datenschutzrichtlinie für die lokale Warteschlange auf BNK7 zu definieren:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r CN=Fi
nAdm2,O=BCO,C=US
```

In dieser Richtlinie wird der WS-Manager als BNK7 identifiziert. Der Richtlinienname und die zugehörige Warteschlange sind FIN.RCPT.Q7. Der für die Signatur des Absenders erwartete Algorithmus ist SHA1, der definierte Name (DN) des sendenden Benutzers soll 'CN=Teller5, O=BCO, C=US' sein, und der Empfängerbenutzer ist 'CN=FinAdm2, O=BCO, C=US'. Der Algorithmus, der zum Entschlüsseln der Nachrichtendaten verwendet wird, ist 3DES.

Nach dem Definieren der beiden Richtlinien starten Sie die Warteschlangenmanager BNK6 und BNK7 erneut oder verwenden den z/OS-Befehl **MODIFY**, um die Advanced Message Security-Richtlinienkonfiguration zu aktualisieren. For example:

Auf BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

Auf BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

Leitfaden für den Schnelleinstieg für AMS mit Java-Clients

Verwenden Sie dieses Handbuch für die schnelle Konfiguration von Advanced Message Security, um die Nachrichtensicherheit für Java-Anwendungen bereitzustellen, die eine Verbindung mithilfe von Clientbindungen herstellen. Wenn Sie die Operation abgeschlossen haben, haben Sie einen Schlüsselspeicher erstellt, um Benutzeridentitäten und definierte Richtlinien für die Signierung/Verschlüsselung für Ihren Warteschlangenmanager zu prüfen.

Vorbereitende Schritte

Stellen Sie sicher, dass die entsprechenden Komponenten installiert sind, wie im **Leitfaden für den Schnelleinstieg** (Windows oder AIX and Linux) beschrieben ist.

1. WS-Manager und eine Warteschlange erstellen

Informationen zu diesem Vorgang

In allen folgenden Beispielen wird eine Warteschlange mit dem Namen TEST.Q verwendet, um Nachrichten zwischen Anwendungen zu übergeben. Advanced Message Security verwendet Interceptors, um

Nachrichten an dem Punkt zu signieren und zu verschlüsseln, an dem Sie in der IBM MQ-Infrastruktur über die Standardschnittstelle von IBM MQ eintreffen. Die Basiseinrichtung wird in IBM MQ vorgenommen und in den folgenden Schritten konfiguriert.

Vorgehensweise

1. Einen WS-Manager erstellen

```
crtmqm QM_VERIFY_AMS
```

2. WS-Manager starten

```
strmqm QM_VERIFY_AMS
```

3. Erstellen und starten Sie einen Listener, indem Sie die folgenden Befehle in **runmqsc** für Warteschlangenmanager QM_VERIFY_AMS eingeben.

```
DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
START LISTENER(AMS.LSTR)
```

4. Erstellen Sie einen Kanal, über den die Anwendungen eine Verbindung herstellen können, indem Sie folgenden Befehl in **runmqsc** für Warteschlangenmanager QM_VERIFY_AMS eingeben:

```
DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
```

5. Erstellen Sie eine Warteschlange mit dem Namen TEST.Q , indem Sie den folgenden Befehl in **runmqsc** für Warteschlangenmanager QM_VERIFY_AMS eingeben.

```
DEFINE QLOCAL(TEST.Q)
```

Ergebnisse

Wenn die Prozedur erfolgreich ausgeführt wurde, zeigt der folgende in **runmqsc** eingegebene Befehl Details zu TEST.Qan:

```
DISPLAY Q(TEST.Q)
```

2. Benutzer erstellen und berechtigen

Informationen zu diesem Vorgang

In diesem Szenario werden zwei Benutzer angezeigt: alice, der Sender und bob, der Empfänger. Um die Anwendungswarteschlange verwenden zu können, müssen diese Benutzer berechtigt sein, sie zu verwenden. Um auch die in diesem Szenario definierten Zugriffsschutzrichtlinien erfolgreich zu verwenden, müssen diesen Benutzern Zugriff auf einige Systemwarteschlangen erteilt werden. Weitere Informationen zum Befehl **setmqaut** finden Sie unter **setmqaut**.

Vorgehensweise

1. Erstellen Sie zwei Benutzer, wie im **Leitfaden für den Schnelleinstieg** (Windows oder AIX and Linux) für Ihre Plattform beschrieben.
2. Benutzer berechtigen, eine Verbindung zum WS-Manager herzustellen und mit der Warteschlange zu arbeiten

```
setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put
setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get +inq +browse
```

3. Außerdem sollten Sie den beiden Benutzern die Möglichkeit geben, die Systemrichtlinienwarteschlange zu durchsuchen und Nachrichten in die Fehlerwarteschlange einzureihen.

```
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p alice -p bob +browse
setmqaut -m QM_VERIFY_AMS -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p alice -p bob +put
```

Achtung: IBM MQ optimiert die Leistung durch das Caching von Richtlinien, sodass Sie keine Datensätze nach Richtliniendetails auf dem SYSTEM.PROTECTION.POLICY.QUEUE in allen Fällen.

Von IBM MQ werden nicht alle verfügbaren Richtlinien zwischengespeichert. Wenn eine hohe Anzahl an Richtlinien vorhanden ist, wird von IBM MQ eine begrenzte Anzahl von Richtlinien zwischengespeichert. Wenn also der Warteschlangenmanager eine geringe Anzahl von Richtlinien definiert hat, ist es nicht erforderlich, die Option zum Durchsuchen für SYSTEM.PROTECTION.POLICY.QUEUE bereitzustellen.

Sie sollten jedoch die Berechtigung zum Durchsuchen dieser Warteschlange erteilen, falls eine hohe Anzahl an Richtlinien definiert ist, oder wenn Sie alte Clients verwenden. Die Warteschlange SYSTEM.PROTECTION.ERROR.QUEUE wird zum Einreihen von Fehlernachrichten verwendet, die vom AMS-Code generiert werden. Die Einreihungsberechtigung für diese Warteschlange wird nur überprüft, wenn Sie versuchen, eine Fehlernachricht in die Warteschlange einzureihen. Ihre Einreihungsberechtigung für die Warteschlange wird nicht überprüft, wenn Sie versuchen, eine Nachricht in eine AMS-geschützte Warteschlange einzureihen oder daraus abzurufen.

Ergebnisse

Die Benutzer werden jetzt erstellt, und die erforderlichen Berechtigungen werden ihnen erteilt.

Nächste Schritte

Um zu überprüfen, ob die Schritte ordnungsgemäß ausgeführt wurden, verwenden Sie die Muster `JmsProducer` und `JmsConsumer` wie im Abschnitt „7. Setup testen” auf Seite 674 beschrieben.

3. Schlüsseldatenbank und Zertifikate erstellen

Informationen zu diesem Vorgang

Um die Nachricht an den Interceptor zu verschlüsseln, muss der öffentliche Schlüssel des sendenden Benutzers verwendet werden. Daher muss die Schlüsseldatenbank der Benutzeridentitäten, die öffentlichen und privaten Schlüsseln zugeordnet sind, erstellt werden. Im realen System, in dem Benutzer und Anwendungen auf mehreren Computern verteilt sind, hat jeder Benutzer seinen eigenen privaten Schlüsselspeicher. Ebenso erstellen wir in diesem Handbuch Schlüsseldatenbanken für `alice` und bob und nutzen die Benutzerzertifikate zwischen ihnen gemeinsam.

Anmerkung: In diesem Handbuch werden Beispielanwendungen verwendet, die in Java geschrieben sind und über Clientbindungen verbunden sind. Wenn Sie Java-Anwendungen mit lokalen Bindungen oder C-Anwendungen verwenden möchten, müssen Sie mit dem Befehl **runmqakm** einen CMS-Keystore und Zertifikate erstellen. Dies wird im **Leitfaden für den Schnelleinstieg** (Windows oder AIX and Linux) dargestellt.

Vorgehensweise

1. Erstellen Sie ein Verzeichnis, in dem Sie Ihren Keystore erstellen können, z. B. `/home/alice/.mqs`. Sie können für die Erstellung möglicherweise das gleiche Verzeichnis verwenden, das im **Leitfaden für den Schnelleinstieg** (Windows oder AIX and Linux) für Ihre Plattform verwendet wird.

   Anmerkung: Dieses Verzeichnis wird in den folgenden Schritten als *keystore-dir* bezeichnet.
2. Erstellen Sie einen neuen Schlüsselspeicher und ein Zertifikat, das den Benutzer `alice` für die Verwendung in der Verschlüsselung identifiziert.

   Anmerkung: Der Befehl **keytool** ist Teil der JRE.

```
keytool -genkey -alias Alice_Java_Cert -keyalg RSA -keystore keystore-dir/keystore.jks -sto
repass passw0rd
-dname "CN=alice, O=IBM, C=GB" -keypass passw0rd
```

Anmerkung:

- Wenn Ihr *keystore-dir* Leerzeichen enthält, müssen Sie den vollständigen Namen Ihres Schlüsselspeichers in Anführungszeichen setzen.
- Es ist ratsam, ein sicheres Kennwort zu verwenden, um den Schlüsselspeicher zu sichern.
- Im Sinne dieses Leitfadens verwenden wir selbst signiertes Zertifikat, das ohne Verwendung einer Zertifizierungsstelle erstellt werden kann. Bei Produktionssystemen ist es ratsam, keine selbst signierten Zertifikate zu verwenden, sondern sich auf Zertifikate zu stützen, die von einer Zertifizierungsstelle signiert wurden.
- Der Parameter **alias** gibt den Namen für das Zertifikat an, in dem die Interceptors nach den erforderlichen Informationen suchen.
- Der Parameter **dname** gibt die Details zu **Definierter Name** (DN) an, die für jeden Benutzer eindeutig sein müssen.

3. Stellen Sie unter AIX and Linux sicher, dass der Keystore gelesen werden kann

```
chmod +r keystore-dir/keystore.jks
```

4. Wiederholen Sie die Schritte 1 bis 4 für Benutzer bob

Ergebnisse

Die beiden Benutzer `alice` und bob verfügen nun jeweils über ein selbst signiertes Zertifikat.

4. Keystore.conf erstellen

Informationen zu diesem Vorgang

Advanced Message Security-Interceptors müssen auf das Verzeichnis verweisen, in dem sich die Schlüsseldatenbanken und Zertifikate befinden. Dies geschieht über die `keystore.conf`-Datei, die diese Informationen im Klartext-Formular enthält. Jeder Benutzer muss über eine separate `keystore.conf`-Datei verfügen. Dieser Schritt sollte sowohl für `alice` als auch für bob ausgeführt werden.

Beispiel

Für dieses Szenario ist der Inhalt von `keystore.conf` für `alice` wie folgt:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Alice_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

Für dieses Szenario ist der Inhalt von `keystore.conf` für bob wie folgt:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Bob_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

Anmerkung:

- Der Pfad zu der Keystore-Datei muss ohne Dateierweiterung angegeben werden.

- Wenn Sie bereits über eine `keystore.conf`-Datei verfügen, weil Sie die Anweisungen im Handbuch für den Schnelleinstieg (Windows oder AIX and Linux) verfolgt haben, können Sie die vorhandene Datei bearbeiten, um diese Zeilen hinzuzufügen.
- Weitere Informationen finden Sie unter „Struktur der Keystore-Konfigurationsdatei (keystore.conf) für AMS” auf Seite 684.

5. Zertifikate gemeinsam nutzen

Informationen zu diesem Vorgang

Geben Sie die Zertifikate zwischen den beiden Keystores frei, so dass jeder Benutzer die andere identifizieren kann. Dies wird durch Extrahieren der einzelnen Benutzerzertifikaten und Importieren in den Schlüsselspeicher des anderen Benutzers erreicht.

Anmerkung: Die Begriffe *extract* und *export* werden von verschiedenen Zertifikatstools unterschiedlich verwendet. Das IBM GSKit **`strmqikm`** -Befehlstool (ikeyman) unterscheidet beispielsweise, dass Sie Zertifikate (öffentliche Schlüssel) *extrahieren* und private Schlüssel *exportieren* . Diese Unterscheidung ist für Tools, die beide Optionen anbieten, extrem wichtig, da die Verwendung von *export* versehentlich Ihre Anwendung durch die Übergabe des privaten Schlüssels vollständig beeinträchtigen würde. Da diese Unterscheidung so wichtig ist, wird in der IBM MQ-Dokumentation darauf geachtet, diese Ausdrücke durchgängig zu verwenden. Im Java-Keytool wird jedoch eine Befehlszeilenoption mit der Bezeichnung *exportcert* bereitgestellt, mit der nur der öffentliche Schlüssel extrahiert wird. Aus diesen Gründen bezieht sich die folgende Prozedur auf das *Extrahieren* von Zertifikaten mithilfe der Option *exportcert* .

Vorgehensweise

1. Extrahieren Sie das Zertifikat, das `alice` identifiziert.

   ```
   keytool -exportcert -keystore alice-keystore-dir/keystore.jks -storepass passw0rd
   -alias Alice_Java_Cert -file alice-keystore-dir/Alice_Java_Cert.cer
   ```

2. Importieren Sie das Zertifikat, das `alice` identifiziert, in den Schlüsselspeicher, den bob verwenden wird. Wenn Sie gefragt werden, ob Sie diesem Zertifikat vertrauen.

   ```
   keytool -importcert -file alice-keystore-dir/Alice_Java_Cert.cer -alias Alice_Java_Cert
   -keystore bob-keystore-dir/keystore.jks -storepass passw0rd
   ```

3. Wiederholen Sie die Schritte für bob.

Ergebnisse

Die beiden Benutzer `alice` und bob sind jetzt in der Lage, einander erfolgreich zu identifizieren, wenn sie selbst signierte und gemeinsam signierte Zertifikate erstellt haben.

Nächste Schritte

Stellen Sie sicher, dass sich ein Zertifikat im Keystore befindet, indem Sie die folgenden Befehle ausführen, die die zugehörigen Details ausgeben:

```
keytool -list -keystore bob-keystore-dir/keystore.jks -storepass passw0rd -alias Alice_Java_Cert
keytool -list -keystore alice-keystore-dir/keystore.jks -storepass passw0rd -alias Bob_Java_Cert
```

6. Warteschlangenrichtlinie definieren

Informationen zu diesem Vorgang

Wenn der Warteschlangenmanager erstellt und die Abfangprozesse für das Abfangen von Nachrichten und den Zugriff auf Verschlüsselungsschlüssel vorbereitet sind, können wir mit dem Befehl `setmqspl` mit dem Definieren von Zugriffsschutzrichtlinien für `QM_VERIFY_AMS` beginnen. Weitere Informationen

zu diesem Befehl finden Sie in setmqspl . Jeder Richtlinienname muss mit dem Namen der Warteschlange identisch sein, auf die er angewendet werden soll.

Beispiel

Dies ist ein Beispiel für eine Richtlinie, die in der TEST.Q -Warteschlange definiert ist, die vom Benutzer `alice` mit dem SHA1 -Algorithmus signiert und mit dem 256-Bit-Algorithmus AES für den Benutzer bob verschlüsselt wurde:

```
setmqspl -m QM_VERIFY_AMS -p TEST.Q -s SHA1 -a "CN=alice,O=IBM,C=GB" -e AES256 -r
"CN=bob,O=IBM,C=GB"
```

Anmerkung: Die DNs stimmen exakt mit den DNs überein, die im Zertifikat des jeweiligen Benutzers aus der Schlüsseldatenbank angegeben sind.

Nächste Schritte

Geben Sie den folgenden Befehl aus, um die von Ihnen definierte Richtlinie zu überprüfen:

```
dspmqspl -m QM_VERIFY_AMS
```

Wenn Sie die Richtliniendetails als Gruppe von `setmqspl` -Befehlen drucken möchten, müssen Sie die Markierung `-export` verwenden. Auf diese Weise können bereits definierte Richtlinien gespeichert werden:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Setup testen

Vorbereitende Schritte

Stellen Sie sicher, dass die von Ihnen installierte Version von Java die unbeschränkten JCE-Richtliniendateien installiert hat.

Anmerkung: Die Version von Java, die in der IBM MQ-Installation bereitgestellt wird, verfügt bereits über diese Richtliniendateien. Sie kann in *MQ_INSTALLATION_PATH*`/java/bin` gefunden werden.

Informationen zu diesem Vorgang

Wenn Sie verschiedene Programme unter verschiedenen Benutzern ausführen, können Sie überprüfen, ob die Anwendung ordnungsgemäß konfiguriert wurde. Informationen zur Ausführung von Programmen mit unterschiedlichen Benutzern finden Sie im **Leitfaden für den Schnelleinstieg** (Windows oder AIX) für Ihre Plattform.

Vorgehensweise

1. Wenn Sie diese JMS-Beispielanwendungen ausführen möchten, verwenden Sie die Einstellung CLASSPATH für Ihre Plattform, wie in Von IBM MQ classes for JMS verwendete Umgebungsvariablen gezeigt, um sicherzustellen, dass das Beispielverzeichnis enthalten ist.
2. Geben Sie als Benutzer `alice` eine Nachricht mit einer Beispielanwendung ein, die als Client eine Verbindung herstellen soll:

   ```
   java JmsProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

3. Als Benutzer bob eine Nachricht mit einer Beispielanwendung abrufen, die als Client eine Verbindung herstellen soll:

   ```
   java JmsConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

Ergebnisse

Wenn die Anwendung für beide Benutzer ordnungsgemäß konfiguriert wurde, wird die Nachricht des Benutzers `alice` angezeigt, wenn bob die Anwendung "Erholen" ausführt.

Remote-Warteschlangen unter AMS schützen

Um ferne Warteschlangen vollständig zu schützen, müssen die Richtlinien in der fernen Warteschlange und in der lokalen Warteschlange festgelegt werden, an die Nachrichten übertragen werden.

Wenn eine Nachricht in eine ferne Warteschlange eingereiht wird, fängt Advanced Message Security die Operation ab und verarbeitet die Nachricht gemäß einer Richtlinie, die für die ferne Warteschlange festgelegt ist. Für eine Verschlüsselungsrichtlinie wird die Nachricht beispielsweise verschlüsselt, bevor sie zur Verarbeitung an IBM MQ übergeben wird. Nachdem Advanced Message Security die Nachricht verarbeitet hat, die in eine ferne Warteschlange eingereiht wurde, stellt IBM MQ sie in die zugehörige Übertragungswarteschlange und leitet sie an den Zielwarteschlangenmanager und die Zielwarteschlange weiter.

Wenn eine GET-Operation in der lokalen Warteschlange ausgeführt wird, versucht Advanced Message Security, die Nachricht gemäß der Richtlinie zu entschlüsseln, die in der lokalen Warteschlange festgelegt ist. Damit die Operation erfolgreich ist, muss die Richtlinie, die zum Entschlüsseln der Nachricht verwendet wird, mit der für die Verschlüsselung verwendeten Richtlinie identisch sein. Jede Diskrepanz führt dazu, dass die Nachricht zurückgewiesen wird.

Wenn aus irgendeinem Grund nicht beide Richtlinien gleichzeitig definiert werden können, wird eine stufenweise Rollout-Unterstützung bereitgestellt. Die Richtlinie kann in einer lokalen Warteschlange mit der Toleranzmarkierung gesetzt werden, die angibt, dass eine Richtlinie, die einer Warteschlange zugeordnet ist, ignoriert werden kann, wenn ein Versuch, eine Nachricht aus der Warteschlange abzurufen, eine Nachricht enthält, für die der Sicherheitsrichtliniensatz nicht definiert ist. In diesem Fall versucht GET, die Nachricht zu entschlüsseln, aber es ist möglich, dass nicht verschlüsselte Nachrichten zugestellt werden. Auf diese Weise können Richtlinien für ferne Warteschlangen festgelegt werden, nachdem die lokalen Warteschlangen geschützt (und getestet) wurden.

Hinweis: Entfernen Sie das Toleranz-Flag, sobald das Advanced Message Security-Rollout abgeschlossen ist.

Zugehörige Verweise

setmqspl (Sicherheitsrichtlinie festlegen)

Routing geschützter Nachrichten unter AMS mit IBM Integration Bus

Advanced Message Security kann Nachrichten in einer Infrastruktur schützen, in der IBM Integration Bus oder WebSphere Message Broker 8.0.0.1 (oder höher) installiert ist. Sie sollten die Spezifik beider Produkte verstehen, bevor Sie die Sicherheit in der IBM Integration Bus-Umgebung anwenden.

Informationen zu diesem Vorgang

Advanced Message Security stellt eine umfassende Sicherheit für die Nachrichtennutzdaten bereit. Dies bedeutet, dass nur die Parteien, die als die gültigen Absender und Empfänger einer Nachricht angegeben sind, in der Lage sind, sie zu erzeugen oder zu empfangen. Dies impliziert, dass Sie zur Sicherung von Nachrichten, die durch den IBM Integration Bus geleitet werden, den IBM Integration Bus berechtigen, Nachrichten ohne Kenntnisse der entsprechenden Inhalte zu verarbeiten (Szenario 1) oder ihn als berechtigten Benutzer festlegen, der Nachrichten empfangen und senden kann (Szenario 2).

Szenario 1-Der Integration Bus kann keinen Nachrichteninhalt anzeigen.

Vorbereitende Schritte

Ihr IBM Integration Bus sollte mit einem vorhandenen Warteschlangenmanager verbunden sein. Ersetzen Sie *QMgrName* durch diesen vorhandenen WS-Manager-Namen in den folgenden Befehlen.

Informationen zu diesem Vorgang

In diesem Szenario stellt Alice eine geschützte Nachricht in eine Eingabewarteschlange QIN. Basierend auf der Nachrichteneigenschaft routeTowird die Nachricht entweder an *bob* (QBOB) weitergeleitet.[1](QCECIL) oder die Standardwarteschlange (QDEF). Die Weiterleitung ist möglich, da Advanced Message Security nur die Nachrichtennutzdaten schützt, nicht jedoch die zugehörigen Header und Eigenschaften, die ungeschützt bleiben und von IBM Integration Bus gelesen werden können. Advanced Message Security wird nur von *alice*, *bob* und *cecil* verwendet. Es muss nicht für den IBM Integration Bus installiert oder konfiguriert werden.

IBM Integration Bus empfängt die geschützte Nachricht aus der ungeschützten Aliaswarteschlange, um jeden Versuch zum Entschlüsseln der Nachricht zu vermeiden. Wenn die geschützte Warteschlange direkt verwendet werden sollte, wird die Nachricht in die Warteschlange DEAD LETTER gestellt, die nicht entschlüsselt werden kann. Die Nachricht wird vom IBM Integration Bus weitergeleitet und kommt unverändert in der Zielwarteschlange an. Daher wird sie immer noch vom ursprünglichen Autor signiert (sowohl *bob* als auch *cecil*) akzeptieren nur Nachrichten, die von *alice* gesendet wurden) und wie zuvor geschützt (nur *bob* und *cecil* können es lesen). IBM Integration Bus reiht die weitergeleitete Nachricht in eine ungeschützte Aliaswarteschlange ein. Die Empfänger rufen die Nachricht aus einer geschützten Ausgabewarteschlange ab, wo sie von AMS transparent entschlüsselt wird.

Vorgehensweise

1. Konfigurieren Sie *alice*, *bob* und *cecil* zur Verwendung von Advanced Message Security, wie im **Leitfaden für den Schnelleinstieg** (Windows oder AIX) beschrieben.

   Stellen Sie sicher, dass die folgenden Schritte ausgeführt werden:

   - Benutzer erstellen und berechtigen
   - Schlüsseldatenbank und Zertifikate erstellen
   - Keystore.conf wird erstellt
2. Geben Sie *alice* das Zertifikat *bob* und *cecil* an, sodass *alice* bei der Überprüfung von digitalen Signaturen in Nachrichten von ihnen identifiziert werden kann.

   Führen Sie dazu das Zertifikat aus, das *alice* für eine externe Datei identifiziert, und fügen Sie anschließend das extrahierte Zertifikat den Keystores *bob* und *cecil* hinzu. Es ist wichtig, dass Sie die in **Aufgabe 5 angegebene Methode verwenden. Gemeinsame Nutzung von Zertifikaten** im **Leitfaden für den Schnelleinstieg** (Windows oder AIX).
3. Geben Sie *bob* und *cecil* Zertifikate an *alice* an, sodass *alice* Nachrichten, die für *bob* und *cecil* verschlüsselt sind, senden kann.

   Verwenden Sie dazu die im vorherigen Schritt angegebene Methode.
4. Definieren Sie in Ihrem Warteschlangenmanager die lokalen Warteschlangen mit dem Namen QIN, QBOB, QCECIL und QDEF.

   ```
   DEFINE QLOCAL(QIN)
   ```
5. Richten Sie die Sicherheitsrichtlinie für die QIN -Warteschlange in eine auswählbare Konfiguration ein. Verwenden Sie die identische Konfiguration für die Warteschlangen QBOB, QCECIL und QDEF .

   ```
   setmqspl -m QMgrName -p QIN -s SHA1 -a "CN=alice,O=IBM,C=GB"
   -e AES256 -r "CN=bob,O=IBM,C=GB" -r "CN=cecil,O=IBM,C=GB"
   ```

   In diesem Szenario wird die Sicherheitsrichtlinie vorausgesetzt, bei der *alice* der einzige berechtigte Absender ist und *bob* und *cecil* die Empfänger sind.
6. Definieren Sie Aliaswarteschlangen AIN, ABOB und ACECIL , die die lokalen Warteschlangen QIN, QBOB bzw. QCECIL referenzieren.

[1] Cecil's

```
DEFINE QALIAS(AIN) TARGET(QIN)
```

7. Stellen Sie sicher, dass die Sicherheitskonfiguration für die im vorherigen Schritt angegebenen Aliasnamen nicht vorhanden ist. Andernfalls wird die zugehörige Richtlinie auf NONE gesetzt.

```
dspmqspl -m QMgrName -p AIN
```

8. Erstellen Sie in IBM Integration Bus einen Nachrichtenfluss, um die Nachrichten weiterzuleiten, die in der Aliaswarteschlange AIN für den BOB-, CECIL- oder DEF-Knoten eingehen, je nach der Eigenschaft routeTo der Nachricht. Um dies zu tun:
   a) Erstellen Sie einen MQInput -Knoten mit dem Namen IN und ordnen Sie den Aliasnamen AIN als Warteschlangennamen zu.
   b) Erstellen Sie MQOutput -Knoten mit dem Namen BOB, CECIL und DEF, und ordnen Sie Aliaswarteschlangen ABOB, ACECIL und ADEF als ihre jeweiligen Warteschlangennamen zu.
   c) Erstellen Sie einen Routenknoten und rufen Sie ihn TEST auf.
   d) Verbinden Sie den IN -Knoten mit dem Eingabeterminal des TEST -Knotens.
   e) Erstellen Sie bob-und cecil -Ausgabeterminals für den TEST -Knoten.
   f) Verbinden Sie das bob -Ausgabeterminal mit dem BOB -Knoten.
   g) Verbinden Sie das cecil -Ausgabeterminal mit dem CECIL -Knoten.
   h) Verbinden Sie den DEF-Knoten mit dem Standardausgabeterminal.
   i) Wenden Sie die folgenden Regeln an:

```
$Root/MQRFH2/usr/routeTo/text()="bob"
$Root/MQRFH2/usr/routeTo/text()="cecil"
```

9. Implementieren Sie den Nachrichtenfluss in der Laufzeitkomponente für den IBM Integration Bus.
10. Wird als Benutzer Alice ausgeführt, wird eine Nachricht ausgegeben, die auch eine Nachrichteneigenschaft mit dem Namen routeTo mit dem Wert bob oder cecil enthält. Wenn Sie die Beispielanwendung **amqsstm** ausführen, können Sie dies tun.

```
Sample AMQSSTMA start
target queue is TEST.Q
Enter property name
routeTo
Enter property value
bob
Enter property name

Enter message text
My Message to Bob
Sample AMQSSTMA end
```

11. Bei der Ausführung als Benutzer *bob* wird die Nachricht aus der Warteschlange QBOB mithilfe der Beispielanwendung **amqsget**abgerufen.

Ergebnisse

Wenn *alice* eine Nachricht in die QIN -Warteschlange einreiht, wird die Nachricht geschützt. Sie wird in geschützter Form durch den IBM Integration Bus aus der Aliaswarteschlange AIN abgerufen. Der IBM Integration Bus legt fest, wohin die Nachricht weitergeleitet werden soll, die die Eigenschaft routeTo liest, die wie alle Eigenschaften nicht verschlüsselt ist. IBM Integration Bus reiht die Nachricht in die entsprechende ungeschützte Aliaswarteschlange ein, wodurch ein weiterer Schutz verhindert wird. Wird die Nachricht von *bob* oder *cecil* aus der Warteschlange empfangen, wird die Nachricht entschlüsselt und die digitale Signatur geprüft.

Informationen zu diesem Vorgang

In diesem Szenario ist eine Gruppe von Einzelpersonen berechtigt, Nachrichten an den IBM Integration Bus zu senden. Eine weitere Gruppe kann Nachrichten empfangen, die vom IBM Integration Bus erstellt werden. Die Übertragung zwischen den Parteien und dem IBM Integration Bus kann nicht abgehört werden.

Beachten Sie, dass der IBM Integration Bus Schutzrichtlinien und Zertifikate nur liest, wenn eine Warteschlange geöffnet ist. Daher müssen Sie die Ausführungsgruppe nach jeder Aktualisierung erneut laden, damit Schutzrichtlinien für die Änderungen wirksam werden.

```
mqsireload execution-group-name
```

Wenn IBM Integration Bus als berechtigter Teilnehmer betrachtet wird, der die Nachrichtennutzdaten lesen oder signieren kann, müssen Sie Advanced Message Security für den Benutzer konfigurieren, der den IBM Integration Bus-Service startet. Beachten Sie, dass es sich dabei nicht unbedingt um denselben Benutzer handelt, der Nachrichten in Warteschlangen einreiht oder von dort abruft, oder um den Benutzer, der die IBM Integration Bus-Anwendungen erstellt und implementiert.

Vorgehensweise

1. Konfigurieren Sie *alice*, *bob*, *cecil* und *dave* sowie den IBM Integration Bus-Servicebenutzer zur Verwendung von Advanced Message Security, wie im **Leitfaden für den Schnelleinstieg** (Windows oder AIX) beschrieben.

   Stellen Sie sicher, dass die folgenden Schritte ausgeführt werden:

   - Benutzer erstellen und berechtigen
   - Schlüsseldatenbank und Zertifikate erstellen
   - Keystore.conf wird erstellt
2. Stellen Sie die Zertifikate von *alice*, *bob*, *cecil* und *dave* dem Servicebenutzer von IBM Integration Bus bereit.

   Extrahieren Sie dazu alle Zertifikate, mit denen *alice*, *bob*, *cecil* und *dave* angegeben werden, in externe Dateien und fügen Sie die extrahierten Zertifikate anschließend dem IBM Integration Bus-Keystore hinzu. Es ist wichtig, dass Sie die in **Aufgabe 5 angegebene Methode verwenden. Gemeinsame Nutzung von Zertifikaten** im **Leitfaden für den Schnelleinstieg** (Windows oder AIX).
3. Stellen Sie das Zertifikat des IBM Integration Bus Servicebenutzers für *alice*, *bob*, *cecil* und *dave*bereit.

   Verwenden Sie dazu die im vorherigen Schritt angegebene Methode.

   Anmerkung: *Alice* und *bob* benötigen das Zertifikat des IBM Integration Bus-Servicebenutzers, um die Nachrichten korrekt verschlüsseln zu können. Der IBM Integration Bus-Servicebenutzer benötigt die Zertifikate von *alice* und *bob*, um Autoren der Nachrichten prüfen zu können. Der IBM Integration Bus-Servicebenutzer benötigt die Zertifikate von *cecil* und *dave*, um Nachrichten für diese verschlüsseln zu können. *cecil* und *dave* benötigen die Zertifikate des IBM Integration Bus-Servicebenutzers, um zu prüfen, ob die Nachricht vom IBM Integration Bus stammt.
4. Definieren Sie eine lokale Warteschlange mit der Bezeichnung `IN` und definieren Sie die Sicherheitsrichtlinie mit *alice* und *bob*, die als Autoren angegeben sind, und den Servicebenutzer für den IBM Integration Bus, der als Empfänger angegeben ist:

   ```
   setmqspl -m QMgrName -p IN -s MD5 -a "CN=alice,O=IBM,C=GB" -a "CN=bob,O=IBM,C=GB"
   -e AES256 -r "CN=broker,O=IBM,C=GB"
   ```
5. Definieren Sie eine lokale Warteschlange mit der Bezeichnung `OUT` und definieren Sie die Sicherheitsrichtlinie mit dem Servicebenutzer für den IBM Integration Bus, der als Autor angegeben ist, und *cecil* und *dave*, die als Empfänger angegeben sind:

```
setmqspl -m QMgrName -p OUT -s MD5 -a "CN=broker,O=IBM,C=GB" -e AES256
-r "CN=cecil,O=IBM,C=GB" -r "CN=dave,O=IBM,C=GB"
```

6. Erstellen Sie in IBM Integration Bus einen Nachrichtenfluss mit den Knoten `MQInput` und `MQOutput`. Konfigurieren Sie den `MQInput` -Knoten für die Verwendung der `IN` -Warteschlange und des `MQOutput` -Knotens, um die `OUT` -Warteschlange zu verwenden.
7. Implementieren Sie den Nachrichtenfluss in der Laufzeitkomponente für den IBM Integration Bus.
8. Bei Ausführung als Benutzer *alice* oder *bob* wird eine Nachricht mithilfe der Beispielanwendung `IN` in die Warteschlange eingereiht **amqsput**.
9. Bei der Ausführung als Benutzer *cecil* oder *dave* wird die Nachricht `OUT` mithilfe der Beispielanwendung **amqsget**aus der Warteschlange abgerufen.

Ergebnisse

Nachrichten, die von *alice* oder *bob* an die Eingabewarteschlange `IN` gesendet werden, sind verschlüsselt und können nur vom IBM Integration Bus gelesen werden. IBM Integration Bus akzeptiert nur Nachrichten von *alice* und *bob* und lehnt alle anderen ab. Die akzeptierten Nachrichten werden entsprechend verarbeitet, signiert und mit den Schlüsseln *cecil* und *dave' s* verschlüsselt, bevor sie in die Ausgabewarteschlange `OUT` gestellt werden. Nur *cecil* und *dave* können die Nachricht lesen, während Nachrichten, die nicht vom IBM Integration Bus signiert sind, abgelehnt werden.

Advanced Message Security mit Managed File Transfer verwenden

In diesem Szenario wird erläutert, wie Advanced Message Security konfiguriert wird, um die Vertraulichkeit von Nachrichten für Daten bereitzustellen, die über Managed File Transfer gesendet werden.

Vorbereitende Schritte

Stellen Sie sicher, dass die Advanced Message Security-Komponente auf der IBM MQ-Installation installiert ist, auf der sich die von Managed File Transfer verwendeten Warteschlangen befinden, die Sie schützen möchten.

Wenn Ihre Managed File Transfer-Agenten eine Verbindung im Bindungsmodus herstellen, stellen Sie sicher, dass auch die GSKit-Komponente in der zugehörigen lokalen Installation installiert ist.

Informationen zu diesem Vorgang

Wenn die Datenübertragung zwischen zwei Managed File Transfer-Agenten unterbrochen ist, verbleiben möglicherweise vertrauliche Daten ungeschützt in den zu Grunde liegenden IBM MQ-Warteschlangen, mit denen die Übertragung verwaltet wird. In diesem Szenario wird erläutert, wie Advanced Message Security konfiguriert und verwendet wird, um solche Daten in den Managed File Transfer-Warteschlangen zu schützen.

In diesem Szenario wird eine einfache Topologie betrachtet, die eine Maschine mit zwei Managed File Transfer -Warteschlangen und zwei Agenten (`AGENT1` und `AGENT2`) umfasst, die einen einzelnen Warteschlangenmanager gemeinsam nutzen, wie im Szenario Managed File Transfer Szenariobeschrieben. Beide Agenten verbinden sich auf die gleiche Weise, entweder im Bindungsmodus oder im Clientmodus.

1. Zertifikate erstellen

Vorbereitende Schritte

In diesem Szenario wird ein einfaches Modell verwendet, in dem Benutzer `ftagent` in der Gruppe `FTAGENTS` für die Ausführung der Managed File Transfer Agent-Prozesse verwendet wird. Wenn Sie Ihre eigenen Benutzer-und Gruppennamen verwenden, ändern Sie die Befehle entsprechend.

Informationen zu diesem Vorgang

Advanced Message Security verwendet die Verschlüsselung mit öffentlichen Schlüsseln, um Nachrichten in geschützten Warteschlange zu signieren und/oder zu verschlüsseln.

Anmerkung:

- Wenn Ihre Managed File Transfer-Agenten im Bindungsmodus ausgeführt werden, finden Sie die Befehle, die Sie zur Erstellung eines CMS-Keystores (Cryptographic Message Syntax) verwenden müssen, im **Leitfaden für den Schnelleinstieg** (Windows oder AIX)für Ihre Plattform.
- Wenn Ihre Managed File Transfer -Agenten im Clientmodus ausgeführt werden, werden die Befehle, die Sie zum Erstellen eines JKS (Java Keystore) benötigen, in „Leitfaden für den Schnelleinstieg für AMS mit Java-Clients” auf Seite 669ausführlich beschrieben.

Vorgehensweise

1. Erstellen Sie ein selbst signiertes Zertifikat, um den Benutzer `ftagent` zu identifizieren, wie in dem entsprechenden Handbuch für den Schnelleinstieg beschrieben.
   Verwenden Sie wie folgt einen definierten Namen (DN):

   ```
   CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>
   ```

2. Erstellen Sie eine `keystore.conf`-Datei, um die Position des Keystores und des Zertifikats innerhalb des Keystores zu identifizieren, wie im entsprechenden Leitfaden für den Schnelleinstieg beschrieben.

2. Nachrichtenschutz konfigurieren

Informationen zu diesem Vorgang

Sie sollten mit dem Befehl **`setmqspl`** eine Sicherheitsrichtlinie für die Datenwarteschlange definieren, die von AGENT2verwendet wird. In diesem Szenario wird derselbe Benutzer verwendet, um beide Agenten zu starten, und deshalb sind der Unterzeichner und der Empfänger-DN identisch und stimmen mit dem generierten Zertifikat überein.

Vorgehensweise

1. Zur Vorbereitung für den Schutz beenden Sie die Managed File Transfer-Agenten mit dem Befehl **`fteStopAgent`**.
2. Erstellen Sie eine Sicherheitsrichtlinie, um die SYSTEM.FTE.DATA.AGENT2-Warteschlange zu schützen.

   ```
   setmqspl -m hubQM -p SYSTEM.FTE.DATA.AGENT2 -s SHA1 -a "CN=ftagent, OU=MFT, O=<organisati
   on>, L=<location>, ST=<state>, C=<country>"
   -e AES128 -r "CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>"
   ```

3. Stellen Sie sicher, dass der Benutzer, der den Managed File Transfer Agent-Prozess ausführt, Zugriff zum Durchsuchen der Systemrichtlinienwarteschlange hat und Nachrichten in die Fehlerwarteschlange einreihen kann.

   ```
   setmqaut -m hubQM -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p ftagent +browse
   setmqaut -m hubQM -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p ftagent +put
   ```

4. Starten Sie Ihre Managed File Transfer-Agenten mit dem Befehl **`fteStartAgent`** erneut.
5. Stellen Sie sicher, dass Ihre Agenten erfolgreich erneut gestartet wurden, indem Sie den Befehl **`fteListAgents`** verwenden und überprüfen, ob sich die Agenten im Status READY befinden.

Ergebnisse

Sie können jetzt Übertragungen von AGENT1 an AGENT2 übergeben, und der Dateiinhalt wird sicher zwischen den beiden Agenten übertragen.

Übersicht über die Installation von Advanced Message Security

Sie können die Advanced Message Security-Komponente auf verschiedenen Plattformen installieren.

Prozedur

- Plattformübergreifende Installation von Advanced Message Security.
- Installieren Sie IBM MQ Advanced for z/OS.
- Installieren Sie IBM MQ Advanced for z/OS Value Unit Edition.

Zugehörige Tasks

Advanced Message Security Deinstallieren

z/OS Prüfung für AMS unter z/OS

Advanced Message Security (AMS) für z/OS stellt eine Möglichkeit zur optionalen Prüfung von Operationen durch Anwendungen in richtliniengeschützten Warteschlangen bereit. Wenn IBM System Management Facility (SMF) aktiviert ist, werden Protokolleinträge für erfolgreiche oder fehlgeschlagene Operationen in richtliniengeschützten Warteschlangen generiert. Zu den überwachten Operationen gehören die MQPUT-, MQPUT1-und MQGET-Operationen.

Die Prüfung ist standardmäßig inaktiviert, Sie können sie jedoch aktivieren, indem Sie _AMS_SMF_TYPE und _AMS_SMF_AUDIT in der konfigurierten Language Environment® _CEE_ENVFILE-Datei für den Adressraum von AMS konfigurieren. Weitere Informationen finden Sie unter Prozeduren für Advanced Message Security erstellen. Die Variable _AMS_SMF_TYPE wird zum Angeben des SMF-Satztyps verwendet und ist eine Zahl zwischen 128 und 255. Ein SMF-Satztyp von 180 ist üblich, ist jedoch nicht obligatorisch. Die Prüfung wird inaktiviert, indem ein Wert von 0 angegeben wird. Die Variable _AMS_SMF_AUDIT konfiguriert, ob Prüfsätze für erfolgreiche Operationen erstellt werden, Operationen, die fehlschlagen oder beides. Die Prüfungsoptionen können auch dynamisch geändert werden, während AMS mit Bedienerbefehlen aktiv ist. Weitere Informationen finden Sie unter Betrieb von Advanced Message Security.

Der SMF-Datensatz wird unter Verwendung von Subtypen definiert, wobei Subtyp 1 ein allgemeines Prüfereignis ist. Der SMF-Datensatz enthält alle Daten, die für die Anforderung, die verarbeitet wird, relevant sind.

Der SMF-Datensatz wird durch das Makro CSQ0KSMF zugeordnet (beachten Sie die Null im Makronamen), die in der Zielbibliothek SCSQMACS bereitgestellt wird. Wenn Sie Datenreduktionsprogramme für SMF-Daten schreiben, können Sie dieses Zuordnungsmakro verwenden, um Unterstützung bei der Entwicklung und Anpassung von SMF-Routinen zur Nachbearbeitung zu erhalten.

In den SMF-Datensätzen, die von Advanced Message Security für z/OS erstellt werden, sind die Daten in Abschnitten organisiert. Der Datensatz besteht aus:

- SMF-Standardheader
- Header-Erweiterung, die durch Advanced Message Security für z/OS definiert wird
- Produktabschnitt
- ein Datenabschnitt

Der Produktabschnitt des SMF-Datensatzes ist immer in den Datensätzen vorhanden, die von Advanced Message Security für z/OS erstellt werden. Der Datenabschnitt variiert je nach Subtyp. Derzeit wird ein Subtyp definiert, so dass ein einzelner Datenabschnitt verwendet wird.

SMF wird im Handbuch z/OS System Management Facilities (SA22-7630) beschrieben. Gültige Satztypen werden im Member SMFPRMxx der Datei PARMLIB Ihres Systems beschrieben. Weitere Informationen finden Sie in der SMF-Dokumentation.

Prüfberichtsgenerator für Advanced Message Security (CSQ0USMF)

Advanced Message Security für z/OS stellt ein Prüfberichtsgeneratortool mit dem Namen CSQ0USMF bereit, das in der Installationsbibliothek SCSQAUTH bereitgestellt wird. Die Beispiel-JCL zur Ausführung des CSQ0USMF-Dienstprogramms CSQ40RSM wird in der Installationsbibliothek SCSQPROC bereitgestellt.

Vor der Ausführung des Dienstprogramms CSQ0USMF müssen die SMF-Datensätze des SMF-Typs 180 aus den SMF-Systemdatensätzen in eine sequenzielle Datei geschrieben werden. Beispiel: Dieser JCL-Code erstellt SMF-Datensätze des Typs 180 aus einer SMF-Datei und überträgt sie an eine Zieldatei:

```
//IFAUDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYSn.MANn.syst,DISP=SHR
//OUTDD1 DD DSN=your.target.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INDD(INDD1,OPTIONS(DUMP))
OUTDD(OUTDD1,TYPE(180))
/*
```

Sie müssen die tatsächlichen Namen der SMF-Dateigruppe, die von Ihrer Installation verwendet werden, überprüfen. Der Zieldatensatz für die gedumpten Datensätze muss ein Satzformat von VBS und eine Satzlänge von 32760 haben.

Anmerkung: Wenn SMF-Protokolldatenströme verwendet werden, müssen Sie das Programm IFASMFDL verwenden, um einen Speicherauszug für einen Protokolldatenstrom auf einen sequenziellen Datensatz auszustellen. In SMF-Datensätze des Typs 116 verarbeiten finden Sie ein Beispiel für die verwendete JCL.

Die Zieldatei kann dann als Eingabe für das Dienstprogramm CSQ0USMF verwendet werden, um einen AMS-Prüfbericht zu erzeugen. Beispiel:

```
//STEP1 EXEC PGM=CSQ0USMF,
// PARM=('/ -SMFTYPE 180 -M qmgr')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//    DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SMFIN DD DSN=your.target.dataset,DISP=SHR
//
```

Das CSQ0USMF-Programm akzeptiert zwei optionale Parameter, die in Tabelle 101 auf Seite 682 aufgeführt werden:

Tabelle 101. CSQ0USMF, optionale Parameter

Parameter	Wert	Beschreibung
SMFTYPE	nnn	Der SMF-Satztyp, der für den Prüfbericht gültig ist. Das Programm CSQ0USMF verwendet nur SMF-Datensätze, die mit dem SMFTYPE-Wert übereinstimmen, wenn der Bericht generiert wird. Wenn Sie SMFTYPE nicht angeben, wird ein Standardwert von 180 verwendet.
M	qmgr	Der Name des IBM MQ-Warteschlangenmanagers, der für den Prüfbericht gültig ist. Wenn Sie den Parameter -M nicht angeben, enthält der Prüfbericht alle Prüfsätze für alle WS-Manager, die in der SMFIN-Datei dargestellt sind.

Keystores und Zertifikate mit AMS verwenden

Um für IBM MQ-Anwendungen einen transparenten Verschlüsselungsschutz bereitzustellen, verwendet Advanced Message Security die Schlüsselspeicherdatei, in der Zertifikate für öffentliche Schlüssel und private Schlüssel gespeichert werden. Unter z/OS wird anstelle einer Schlüsselspeicherdatei ein SAF-Schlüsselring verwendet.

In Advanced Message Security werden Benutzer und Anwendungen durch PKI-Identitäten (Public Key Infrastructure) dargestellt. Dieser Typ von Identität wird zum Signieren und Verschlüsseln von Nachrichten verwendet. Die PKI-Identität wird durch das Feld **Definierter Name (DN)** des Subjekts in einem Zertifikat dargestellt, das signierten und verschlüsselten Nachrichten zugeordnet ist. Damit ein Benutzer oder

eine Anwendung ihre Nachrichten verschlüsseln kann, müssen sie Zugriff auf die Schlüsselspeicherdatei haben, in der Zertifikate und die zugehörigen privaten und öffentlichen Schlüssel gespeichert werden.

Unter AIX, Linux, and Windows wird die Position des Keystores in der Keystore-Konfigurationsdatei bereitgestellt, die standardmäßig `keystore.conf` ist. Jeder Advanced Message Security-Benutzer muss über die Schlüsselspeicherkonfigurationsdatei verfügen, die auf eine Schlüsselspeicherdatei verweist. Advanced Message Security akzeptiert das folgende Format von Schlüsselspeicher-Dateien: `.kdb`, `.jceks`, `.jks`.

Die Standardposition der `keystore.conf`-Datei lautet wie folgt:

- IBM i Linux AIX Unter IBM i, AIX and Linux: `$HOME/.mqs/keystore.conf`
- Windows Unter Windows: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

  Anmerkung: Der Pfad aufWindows kann und sollte den Laufwerksbuchstaben angeben, wenn mehr als ein Laufwerksbuchstabe vorhanden ist.

Wenn Sie einen angegebenen Keystore-Dateinamen und eine angegebene Position verwenden, sollten Sie die folgenden Befehle verwenden:

- Für Java: `java -D`*`MQS_KEYSTORE_CONF=path/filename app_name`*
- Für C Client und Server:
  - Unter AIX and Linux: `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - Unter Windows: `set MQS_KEYSTORE_CONF=`*`path\filename`*

Sensible Informationen in der `keystore.conf`-Datei schützen

V 9.2.0 V 9.2.0

Für den Zugriff auf sensible Informationen in der Keystore-Datei (z. B Kennwörter) müssen Sie Token bereitstellen, damit IBM MQ Advanced Message Security (AMS) auf den Keystore zugreifen und Nachrichten signieren und verschlüsseln kann.

Sie sollten die sensiblen Informationen in der Konfigurationsdatei des Keystores mit dem Befehl **runamscred** sichern, der mit AMS bereitgestellt wird. Informationen zum Schützen von Konfigurationsdateien finden Sie im Abschnitt „AMS Kennwortschutz für -Konfigurationsdateien einrichten” auf Seite 703.

Beim Schützen von Kennwörtern sollten Sie einen angepassten starken Verschlüsselungsschlüssel verwenden. Für den Zugriff auf Kennwörter während der Ausführung muss dieser Verschlüsselungsschlüssel AMS bereitgestellt werden.

Es gibt zwei Methoden, um die Position der Datei mit dem Verschlüsselungsschlüssel bereitzustellen:

- **amscred.keyfile** Konfigurationseigenschaft in der `keystore.conf` Datei
- **MQS_AMSCRED_KEYFILE**-Umgebungsvariable

Die Vorrangregelung ist **MQS_AMSCRED_KEYFILE**, gefolgt von **amscred.keyfile**und dem Standardschlüssel.

Weitere Informationen finden Sie unter „Advanced Message Security” auf Seite 603.

Zugehörige Konzepte

„Definierte Namen des Senders in AMS” auf Seite 713
Die definierten Namen (DNs) des Absenders identifizieren Benutzer, die berechtigt sind, Nachrichten in eine Warteschlange einzureihen. Ein Absender verwendet sein Zertifikat zum Signieren einer Nachricht, bevor die Nachricht in eine Warteschlange eingereiht wird.

„Definierte Namen des Empfängers in AMS” auf Seite 714
Die definierten Namen (DN) des Empfängers geben Benutzer an, die berechtigt sind, Nachrichten aus einer Warteschlange abzurufen.

Struktur der Keystore-Konfigurationsdatei (keystore.conf) für AMS

Die Keystore-Konfigurationsdatei (`keystore.conf`) verweist Advanced Message Security auf die Position des entsprechenden Keystores.

Jeder der folgenden Konfigurationsdatentypen hat ein Präfix:

V 9.2.0 V 9.2.0 **AMSCRED**
: Parameter, die sich auf das Kennwortschutzsystem beziehen.

CMS
: Certificate Management System, Konfigurationseinträge haben das Präfix `cms.`.

PKCS#11
: Public Key Cryptography Standard #11, Konfigurationseinträge haben das Präfix `pkcs11.`

IBM i **PEM**
: Format Privacy Enhanced Mail, Konfigurationseinträge haben das Präfix `pem.`

JKS
: Java KeyStore, Konfigurationseinträge haben das Präfix `jks.`

JCEKS
: Java Cryptographic Encryption KeyStore, Konfigurationseinträge haben das Präfix `jceks.`

z/OS MQ Adv. VUE **JCERACFKS**
: Java Cryptographic Encryption RACF keyring KeyStore, Konfigurationseinträge haben das Präfix `jceracfks`

Wichtig: Ab IBM MQ 9.0 werden die Werte `JCEKS.provider` und `JKS.provider` ignoriert. Der Bouncy Castle-Provider wird in Verbindung mit der JCE/JCE-Bereitstellung verwendet, die von der verwendeten JRE zur Verfügung gestellt wird. Weitere Informationen finden Sie unter „Unterstützung für Nicht-IBM-JREs mit AMS" auf Seite 688.

Beispielstrukturen für Keystores:

CMS

```
cms.keystore = /dir/keystore_file
cms.certificate = certificate_label
```

PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
```

V 9.2.2 `pkcs11.encrypted = no`

IBM i PEM

```
pem.private = /dir/keystore_file_private_key
pem.public = /dir/keystore_file_public_keys
pem.password = password
```

V 9.2.2 `pem.encrypted = no`

Java JKS

```
jks.keystore = dir/Keystore
jks.certificate = certificate_label
jks.encrypted = no
jks.keystore_pass = password
```

```
jks.key_pass = password
jks.provider = IBMJCE
```

Java JCEKS

```
jceks.keystore = dir/Keystore
jceks.certificate = certificate_label
jceks.encrypted = no
jceks.keystore_pass = password
jceks.key_pass = password
jceks.provider = IBMJCE
```

Java JCERACFKS

```
jceracfks.keystore = safkeyring://user/keyring
jceracfks.certificate = certificate_label
```

Java PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.secondary_keystore_pass = password
pkcs11.encrypted = no
```

Tabelle 102. Zusammenfassung der Parameter, die für die einzelnen Konfigurationsdatentypen erforderlich sind

Parameter	Erforderlich	Konfigurationsdatentyp				
		Java (PKCS#11, JKS, JCEKS und JCE-RACFKS)	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
keystore	✓	✓			✓	
IBM i private	✓		IBM i ✓			
IBM i public	✓		IBM i ✓			
IBM i password	✓		IBM i ✓			
library	✓	✓		✓		
certificate	✓	✓		✓	✓	
token	✓	✓		✓		
token_pin	✓	✓		✓		
secondary_keystore	✓	✓		✓		

Tabelle 102. Zusammenfassung der Parameter, die für die einzelnen Konfigurationsdatentypen erforderlich sind (Forts.)

Parameter	Erforderlich	Konfigurationsdatentyp				
		Java (PKCS#11, JKS, JCEKS und JCERACFKS)	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
`secondary_keystore_password`	✓	✓				
`encrypted`		✓	IBM i V 9.2.2 ✓	V 9.2.2 ✓		
`keystore_pass`	✓	✓				
`key_pass`		✓				
`provider`		✓				
`keyfile`						✓ Sie

Sie können Kommentare mit dem Symbol # hinzufügen.

Konfigurationsdateiparameter werden wie folgt definiert:

keystore
Nur CMS- und Java-Konfiguration.

Pfad zur Schlüsselspeicherdatei für die CMS-, JKS-und JCEKS-Konfiguration.

z/OS MQ Adv. VUE URI zum RACF-Schlüsselring für JCERACFKS-Konfiguration.

Wichtig:

- Der Pfad zu der Keystore-Datei darf die Dateierweiterung nicht enthalten.
- z/OS MQ Adv. VUE Die URI zum RACF-Schlüsselring muss in der folgenden Form vorliegen:

```
safkeyring://user/keyring
```

  Dabei gilt:

  – *user* die Benutzer-ID ist, zu der der Schlüsselring gehört
  – *keyring* der Schlüsselringname ist.

IBM i **private**
Nur PEM-Konfiguration.

Dateiname einer Datei, die den privaten Schlüssel und das Zertifikat im PEM-Format enthält.

IBM i **public**
Nur PEM-Konfiguration.

Dateiname einer Datei, die anerkannte öffentliche Zertifikate im PEM-Format enthält.

IBM i **password**
Nur PEM-Konfiguration.

Kennwort, das zum Entschlüsseln eines verschlüsselten privaten Schlüssels verwendet wird.

V 9.2.2 Sie sollten dieses Feld mit dem nativen AMS-Kennwortschutztool schützen. Weitere Informationen hierzu finden Sie im Abschnitt „Kennwörter schützen” auf Seite 688.

library
Nur PKCS#11.

Pfadname der PKCS#11-Bibliothek.

certificate
Nur CMS-, PKCS#11- und Java-Konfiguration.

Zertifikatsbezeichnung.

token
Nur PKCS#11.

Tokenkennsatz.

token_pin
Nur PKCS#11.

PIN zum Entsperren des Tokens.

V 9.2.0 V 9.2.0 Nur für Java-Operationen; Sie sollten dieses Feld mit dem Tool für den Java AMS-Kennwortschutz schützen. Weitere Informationen dazu finden Sie unter „Kennwörter schützen” auf Seite 688.

V 9.2.2 Nur für native Operationen; Sie sollten dieses Feld mit dem nativen AMS-Kennwortschutztool schützen. Weitere Informationen finden Sie im Abschnitt „Kennwörter schützen” auf Seite 688.

secondary_keystore
Nur PKCS#11.

Der Pfadname des CMS-Keystores, der ohne die Erweiterung ".kdb" bereitgestellt wird und Ankerzertifikate (Stammzertifikate) enthält, die für Zertifikate erforderlich sind, die im PKCS- #11-Token gespeichert sind. Der sekundäre Schlüsselspeicher kann auch Zertifikate enthalten, die in der Trust-Kette enthalten sind, sowie Zertifikate, die in der Datenschutzrichtlinie definiert sind. Dieser CMS-Schlüsselspeicher muss von einer Stashdatei begleitet werden, die sich in demselben Verzeichnis befinden muss wie der sekundäre Schlüsselspeicher.

Für Java -Umgebungen ist ein JKS-Schlüsselspeicher erforderlich und Sie müssen einen **secondary_keystore_password**bereitstellen.

secondary_keystore_password
Nur Java PKCS#11.

Kennwort für den JKS-Keystore, der über die Eigenschaft `secondary_keystore` bereitgestellt wird. Sie sollten dieses Feld mit dem Java AMS-Kennwortschutztool schützen. Weitere Informationen hierzu finden Sie im Abschnitt „Kennwörter schützen” auf Seite 688.

encrypted
V 9.2.0 V 9.2.0 NurJava -Konfiguration.

V 9.2.2 Nur Java-, PKCS#11-und IBM i PEM -Konfiguration.

Status des Kennworts.

keystore_pass
Nur Java-Konfiguration.

Kennwort für die Schlüsselspeicherdatei.

V 9.2.0 V 9.2.0 Nur für Java-Operationen. Sie sollten dieses Feld mit dem Java AMS-Kennwortschutztool schützen. Weitere Informationen hierzu finden Sie im Abschnitt „Kennwörter schützen” auf Seite 688.

key_pass
Nur Java-Konfiguration.

Kennwort für den privaten Schlüssel des Benutzers.

V 9.2.0 V 9.2.0 Nur für Java-Operationen; Sie sollten dieses Feld mit dem Tool für den Java AMS-Kennwortschutz schützen. Weitere Informationen dazu finden Sie unter „Kennwörter schützen” auf Seite 688.

V 9.2.0 V 9.2.0 **keyfile**
Es wird die Position des ursprünglichen Schlüssels angegeben, der beim Schützen oder Entschlüsseln von Kennwörtern in dieser Konfigurationsdatei verwendet wird; siehe „Kennwörter schützen” auf Seite 688

provider
Nur Java-Konfiguration.

Der Java-Sicherheitsprovider, der Verschlüsselungsalgorithmen implementiert, die für das Keystore-Zertifikat erforderlich sind.

Wichtig: Informationen, die im Keystore gespeichert werden, sind für den mit IBM MQ gesendeten sicheren Datenfluss äußerst wichtig. Sicherheitsadministratoren müssen besonders darauf achten, dass sie diesen Dateien Dateiberechtigungen zuordnen.

Kennwörter schützen

V 9.2.0 V 9.2.0

Sie sollten die Kennwörter und andere sensible Informationen, die in der `keystore.conf`-Datei enthalten sind, schützen. Weitere Informationen hierzu finden Sie in **runamscred** .

Beispiel für die `keystore.conf`-Datei:

V 9.2.0 V 9.2.0

```
# Native AMS application configuration
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

# Java AMS application configuration
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

Zugehörige Tasks

„AMS Kennwortschutz für -Konfigurationsdateien einrichten” auf Seite 703
Das Speichern von Kennwörtern für den Keystore und für private Schlüssel als Klartext stellt ein Sicherheitsrisiko dar, weshalb Advanced Message Security ein Tool bereitstellt, das diese Kennwörter mithilfe eines Benutzerschlüssels verschlüsselt.

Unterstützung für Nicht-IBM-JREs mit AMS

IBM MQ classes for Java und IBM MQ classes for JMS unterstützen Advanced Message Security-Operation bei Ausführung mit Nicht-IBM-JREs.

Advanced Message Security (AMS) implementiert Cryptographic Message Syntax (CMS). Die CMS-Syntax wird verwendet, um beliebige Nachrichteninhalte digital zu signieren, zu verdauen, zu authentifizieren oder zu verschlüsseln.

Ab IBM MQ 9.0 nutzt die Advanced Message Security-Unterstützung in IBM MQ classes for Java und IBM MQ classes for JMS die Open-Source-Pakete Bouncy Castle, um CMS zu unterstützen. Das bedeutet, dass diese Klassen die Advanced Message Security-Operation bei Ausführung mit Nicht-IBM-JREs unterstützen.

Vor IBM MQ 9.0 wurde Advanced Message Security nicht in Nicht-IBM-JREs in Java-Clients unterstützt. Advanced Message Security-Unterstützung in IBM MQ classes for Java und IBM MQ classes for JMS hing von der CMS-Unterstützung ab, die von der IBM-Implementierung der Java Cryptography Extensions (JCE) bereitgestellt wurde. Aufgrund dieser Einschränkung war die Funktionalität nur bei Verwendung einer Java runtime environment (JRE) verfügbar, die den JCE-Provider von Java enthielt.

Standort-und Versionsnummerierung für JAR-Dateien von Bouncy Castle

Die Bouncy Castle-JAR-Dateien, die für die Unterstützung von Nicht-IBM-JREs benötigt werden, sind im Rahmen des Installationspakets von IBM MQ classes for Java und IBM MQ classes for JMS enthalten.

Als JAR-Dateien der Bouncy Castle-Datei werden die folgenden Dateien verwendet:

Die Provider-JAR-Datei, die für Bouncy Castle-Operationen von grundlegender Bedeutung ist.
Diese JAR-Datei wird als `bcprov-jdk15on.jar` bezeichnet.

Die JAR-Datei „PKIX“, welche die Unterstützung für CMS-Operationen enthält, die von Advanced Message Security verwendet werden.
Diese JAR-Datei wird als `bcpkix-jdk15on.jar` bezeichnet.

V 9.2.4 V 9.2.0.4 **Die JAR-Datei „util“, die Klassen enthält, die von den anderen Bouncy-Castle-JAR-Dateien verwendet werden.**
Diese JAR-Datei wird als `bcutil-jdk15on.jar` bezeichnet.

Abhängigkeiten

Die IBM MQ 9.1 und spätere Klassen wurden mit IBM-JREs und Oracle-JREs getestet. Sie werden wahrscheinlich auch unter einer beliebigen J2SE-tauglichen JRE erfolgreich ausgeführt. Sie sollten jedoch die folgenden Abhängigkeiten beachten:

- Es sind keine Änderungen an der Advanced Message Security-Konfiguration vorhanden.
- Die Bouncy Castle-Klassen werden nur für CMS-Operationen verwendet. Alle anderen sicherheitsrelevanten Operationen, z. B. der Schlüsselspeicherzugriff, die tatsächliche Verschlüsselung von Daten und die Berechnung von Signaturkontrollsummen, verwenden die Funktionalität, die von der JRE bereitgestellt wird.

  Wichtig: Aus diesem Grund muss die verwendete JRE eine JCE-Providerimplementierung enthalten.
- Wenn Sie einige *starke* Verschlüsselungsalgorithmen verwenden möchten, müssen Sie möglicherweise die *unbeschränkten* Richtliniendateien für die JCE-Implementierung der JRE installieren.

  Weitere Einzelheiten finden Sie in der JRE-Dokumentation.
- Wenn Sie die Java-Sicherheit aktiviert haben:
  - Fügen Sie `java.security.SecurityPermissioninsertProvider.BC` zur Anwendung hinzu, so dass die Bouncy Castle-Klassen als Sicherheitsprovider verwendet werden können.
  - Erteilen Sie `java.security.AllPermission` den JAR-Dateien von Bouncy Castle, welche die Folgenden sind:

    V 9.2.4 V 9.2.0.4 *mq_install_dir*`/java/lib/bcutil-jdk15on.jar`
    mq_install_dir`/java/lib/bcpkix-jdk15on.jar`
    mq_install_dir`/java/lib/bcprov-jdk15on.jar`

Zugehörige Konzepte
Was ist für IBM MQ-Klassen für JMS installiert?
Was ist für IBM MQ-Klassen für Java installiert?

Multi Abfangen des Message Channel Agent (MCA) und AMS

Durch das MCA-Abfangen kann ein Warteschlangenmanager, der unter IBM MQ ausgeführt wird, die für Serververbindungskanäle angewendeten Richtlinien gezielt aktivieren.

Durch das MCA-Abfangen können auch Clients außerhalb von AMS mit einem Warteschlangenmanager verbunden werden und die zugehörigen Nachrichten können verschlüsselt und entschlüsselt werden.

Das MCA-Abfangen soll die AMS-Funktion bereitstellen, wenn AMS nicht als Client aktiviert werden kann. Beachten Sie, dass die Verwendung des MCA-Abfangens und eines AMS-fähigen Clients zu einem doppelten Schutz von Nachrichten führt, was beim Empfang von Anwendungen zu Problemen führen kann. Weitere Informationen finden Sie unter „Advanced Message Security auf dem Client inaktivieren" auf Seite 692.

Anmerkung: MCA-Interceptor werden für AMQP-oder MQTT-Kanäle nicht unterstützt.

Schlüsselspeicherkonfigurationsdatei

Standardmäßig ist die Schlüsselspeicherkonfigurationsdatei für die MCA-Abfangfunktion `key-store.conf` und befindet sich im Verzeichnis `.mqs` im Ausgangsverzeichnispfad des Benutzers, der den Warteschlangenmanager oder den Listener gestartet hat. Der Keystore kann auch unter Verwendung der Umgebungsvariablen MQS_KEYSTORE_CONF konfiguriert werden. Weitere Informationen zur Konfiguration des AMS-Keystores finden Sie unter „Keystores und Zertifikate mit AMS verwenden" auf Seite 682.

Um die MCA-Überwachung zu aktivieren, müssen Sie den Namen eines Kanals angeben, der in der Schlüsselspeicherkonfigurationsdatei verwendet werden soll. Für MCA-Interception kann nur ein Keystore-Typ cms verwendet werden.

Im Abschnitt „MCA-Abfangbeispiel für AMS" auf Seite 690 finden Sie ein Beispiel für die Einrichtung von MCA-Abfangmethoden.

Achtung: Sie müssen die Clientauthentifizierung und die Verschlüsselung auf den ausgewählten Kanälen ausführen, z. B. mit SSL und SSLPEER oder CHLAUTH TYPE (SSLPEERMAP), um sicherzustellen, dass nur berechtigte Clients diese Funktion verbinden und verwenden können.

IBM i

Wenn Ihr Unternehmen IBM iverwendet und Sie eine kommerzielle Zertifizierungsstelle (CA) zum Signieren Ihres Zertifikats ausgewählt haben, erstellt Digital Certificate Manager eine Zertifikatsanforderung im PEM-Format (Privacy-Enhanced Mail). Sie müssen die Anforderung an die von Ihnen ausgewählte Zertifizierungsstelle weiterleiten.

Dazu müssen Sie den folgenden Befehl verwenden, um das richtige Zertifikat für den in `channelname` angegebenen Kanal auszuwählen:

```
pem.certificate.channel.channelname
```

MCA-Abfangbeispiel für AMS

Hier finden Sie eine Beispieltask zur Einrichtung der Überwachung für einen Nachrichtenkanalagenten (Message Chanel Agent, MCA) für AMS.

Vorbereitende Schritte

Achtung: Sie müssen die Clientauthentifizierung und die Verschlüsselung auf den ausgewählten Kanälen ausführen, z. B. mit SSL und SSLPEER oder CHLAUTH TYPE (SSLPEERMAP), um sicherzustellen, dass nur berechtigte Clients diese Funktion verbinden und verwenden können.

Wenn Ihr Unternehmen IBM iverwendet und Sie eine kommerzielle Zertifizierungsstelle (CA) zum Signieren Ihres Zertifikats ausgewählt haben, erstellt Digital Certificate Manager eine Zertifikatsanforderung im PEM-Format (Privacy-Enhanced Mail). Sie müssen die Anforderung an die von Ihnen ausgewählte Zertifizierungsstelle weiterleiten.

Informationen zu diesem Vorgang

Diese Task führt Sie durch den Prozess der Konfiguration Ihres Systems für die Verwendung der MCA-Überwachung und die anschließende Überprüfung der Konfiguration.

Anmerkung: Vor IBM WebSphere MQ 7.5 handelte es sich bei AMS um ein Add-on-Produkt, das separat installiert werden musste und für das Interceptors für den Schutz von Anwendungen konfiguriert wurden. Ab IBM WebSphere MQ 7.5werden die Interceptors automatisch in den MQ-Client- und Serverlaufzeitumgebungen eingeschlossen und dynamisch aktiviert. In diesem Beispiel für die MCA-Überwachung werden die Interceptors am Serverende des Kanals bereitgestellt, und eine ältere Clientlaufzeit wird verwendet (in Schritt 12), um eine ungeschützte Nachricht über den Kanal zu setzen, so dass sie durch die MCA-Interceptor geschützt werden können. Wenn in diesem Beispiel ein Client von IBM WebSphere MQ 7.5 oder höher verwendet wurde, würde dies dazu führen, dass die Nachricht zweimal geschützt werden würde, da der MQ-Client-Runtime-Interceptor und der MCA-Interceptor die Nachricht so schützen würden, wie sie in MQ enthalten ist.

Achtung: Ersetzen Sie `userID` im Code durch Ihre Benutzer-ID.

Vorgehensweise

1. Erstellen Sie die Schlüsseldatenbank und die Zertifikate mit den folgenden Befehlen, um ein Shell-Script zu erstellen.

   Ändern Sie auch **INSTLOC** und **KEYSTORELOC** oder führen Sie die erforderlichen Befehle aus. Beachten Sie, dass Sie das Zertifikat möglicherweise nicht für bob erstellen müssen.

```
INSTLOC=/opt/mq90
KEYSTORELOC=/home/testusr/ssl/ams1
mkdir -p $KEYSTORELOC
chmod -R 777 $KEYSTORELOC
chown -R mqm:mqm $KEYSTORELOC
export PATH=$PATH:$INSTLOC/gskit8/bin
echo "PATH = $PATH"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTLOC/gskit8/lib64

gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd -stash
gsk8capicmd_64 -keydb -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd -stash
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd
-label alice_cert -dn "cn=alice,O=IBM,c=IN" -default_cert yes
gsk8capicmd_64 -cert -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd
-label bob_cert  -dn "cn=bob,O=IBM,c=IN"  -default_cert yes
```

2. Geben Sie die Zertifikate zwischen den beiden Schlüsseldatenbanken frei, so dass jeder Benutzer die andere identifizieren kann.

   Es ist wichtig, dass Sie die in **Aufgabe 5 beschriebene Methode verwenden. Gemeinsame Nutzung von Zertifikaten** im **Leitfaden für den Schnelleinstieg** (Windows oder AIX and Linux).

3. Erstellen Sie `keystore.conf` mit der folgenden Konfiguration: `Keystore.conf location: /home/userID/ssl/ams1/`

```
cms.keystore = /home/userID/ssl/ams1/alicekey
cms.certificate.channel.SYSTEM.DEF.SVRCONN = alice_cert
```

4. WS-Manager `AMSQMGR1` erstellen und starten
5. Definieren Sie einen Listener mit *port* `14567` und *control* `QMGR`
6. Inaktivieren Sie die Kanalberechtigung oder legen Sie die Regeln für die Kanalberechtigung fest.

   Weitere Informationen finden Sie unter SET CHLAUTH .

7. Stoppen Sie den Warteschlangenmanager.
8. Legen Sie den Schlüsselspeicher fest:

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. Starten Sie den WS-Manager auf derselben Shell.
10. Legen Sie die Sicherheitsrichtlinie fest und überprüfen Sie Folgendes:

```
setmqspl -m AMSQMGR1 -s SHA256 -e AES256 -p TESTQ -a "CN=alice,O=IBM,C=IN"
-r "CN=alice,O=IBM,C=IN"
dspmqspl -m AMSQMGR1
```

Weitere Informationen finden Sie in setmqspl und dspmqspl .

11. Legen Sie die Kanalkonfiguration fest:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. Führen Sie **amqsputc** aus einem MQ-Client aus, der einen MCA-Interceptor nicht automatisch aktiviert, z. B. einen Client der IBM WebSphere MQ 7.1 oder früher. Fügen Sie die folgenden beiden Nachrichten ein:

```
/opt/mqm/samp/bin/amqsputc TESTQ TESTQMGR
```

13. Entfernen Sie die Sicherheitsrichtlinie, und überprüfen Sie das Ergebnis:

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

14. Durchsuchen Sie die Warteschlange aus Ihrer IBM MQ 9.0-Installation:

```
/opt/mq90/samp/bin/amqsbcg TESTQ AMSQMGR1
```

Die Durchsuchungsausgabe zeigt die Nachrichten im verschlüsselten Format an.

15. Legen Sie die Sicherheitsrichtlinie fest und überprüfen Sie das Ergebnis:

```
setmqspl -m AMSQMGR1 -s SHA256 -e AES256 -p TESTQ -a "CN=alice,O=IBM,C=IN"
-r "CN=alice,O=IBM,C=IN"
dspmqspl -m AMSQMGR1
```

16. Führen Sie **amqsgetc** aus Ihrer IBM MQ 9.0-Installation aus:

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

Zugehörige Tasks

„Leitfaden für den Schnelleinstieg für AMS mit Java-Clients” auf Seite 669
Verwenden Sie dieses Handbuch für die schnelle Konfiguration von Advanced Message Security, um die Nachrichtensicherheit für Java-Anwendungen bereitzustellen, die eine Verbindung mithilfe von Clientbindungen herstellen. Wenn Sie die Operation abgeschlossen haben, haben Sie einen Schlüsselspeicher erstellt, um Benutzeridentitäten und definierte Richtlinien für die Signierung/Verschlüsselung für Ihren Warteschlangenmanager zu prüfen.

Zugehörige Verweise

„Bekannte Einschränkungen von AMS” auf Seite 640
Es gibt eine Reihe von IBM MQ-Optionen, die nicht unterstützt werden oder Einschränkungen für Advanced Message Security haben.

Advanced Message Security auf dem Client inaktivieren

Sie müssen IBM MQ Advanced Message Security (AMS) inaktivieren, wenn Sie einen IBM WebSphere MQ 7.5 oder einen späteren Client verwenden, um eine Verbindung zu einem Warteschlangenmanager einer früheren Version des Produkts herzustellen und ein 2085 (MQRC_UNKNOWN_OBJECT_NAME)-Fehler gemeldet wird.

Informationen zu diesem Vorgang

Ab IBM WebSphere MQ 7.5 wird IBM MQ Advanced Message Security (AMS) automatisch in einem IBM MQ-Client aktiviert, und deshalb versucht der Client standardmäßig, die Sicherheitsrichtlinien für Objekte im Warteschlangenmanager zu prüfen. Auf Servern früherer Versionen des Produkts, z. B. IBM WebSphere MQ 7.1, ist AMS jedoch nicht aktiviert, was dazu führt, dass der Fehler 2085 (MQRC_UNKNOWN_OBJECT_NAME) gemeldet wird.

Wenn dieser Fehler beim Herstellen einer Verbindung zu einem Warteschlangenmanager aus einer früheren Version des Produkts gemeldet wird, können Sie AMS folgendermaßen inaktivieren:

- Für Java-Clients haben Sie folgende Möglichkeiten:
  - Durch Festlegen einer Umgebungsvariablen AMQ_DISABLE_CLIENT_AMS.
  - Durch Festlegen der Java-Systemeigenschaft com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS.
  - Durch Verwendung der Eigenschaft DisableClientAMS in der Zeilengruppe **Security** in der Datei `mqclient.ini` .
- Für C-Clients auf eine der folgenden Arten:
  - Indem Sie eine Umgebungsvariable MQS_DISABLE_ALL_INTERCEPT setzen.
  - Durch Verwendung der Eigenschaft DisableClientAMS in der Zeilengruppe **Security** in der Datei `mqclient.ini` .

Anmerkung: In IBM WebSphere MQ 7.5 können Sie auch die Umgebungsvariable AMQ_DISABLE_CLIENT_AMS. für C-Clients verwenden. Ab IBM MQ 8.0 kann die Umgebungsvariable AMQ_DISABLE_CLIENT_AMS nicht mehr für C-Clients verwendet werden. Sie müssen stattdessen die Umgebungsvariable MQS_DISABLE_ALL_INTERCEPT verwenden.

Prozedur

- Verwenden Sie zum Inaktivieren von AMS auf dem Client eine der folgenden Optionen:

  AMQ_DISABLE_CLIENT_AMS, Umgebungsvariable
  Sie müssen diese Variable in den folgenden Fällen festlegen:
  - Wenn Sie eine andere Java Runtime Environment (JRE) als IBM Java Runtime Environment (JRE) verwenden
  - Wenn Sie IBM WebSphere MQ 7.5oder höher IBM MQ classes for JMS oder einen IBM MQ classes for Java -Client verwenden.

  Erstellen Sie die Umgebungsvariable AMQ_DISABLE_CLIENT_AMS, und setzen Sie sie auf TRUE in der Umgebung, in der die Anwendung ausgeführt wird. Beispiel:

  ```
  export AMQ_DISABLE_CLIENT_AMS=TRUE
  ```

  Java-Systemeigenschaft com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS
  Für IBM MQ classes for JMS- und IBM MQ classes for Java-Clients können Sie die Java-Systemeigenschaft 'com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS ' auf den Wert TRUE für die Java-Anwendung setzen.

  Sie können beispielsweise die Java-Systemeigenschaft als Option `-D` festlegen, wenn der Java-Befehl aufgerufen wird:

  ```
  java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/java/lib/
  com.ibm.mqjms.jar  my.java.applicationClass
  ```

  Alternativ können Sie die Systemeigenschaft Java in einer Konfigurationsdatei JMS angeben, `jms.config`, wenn die Anwendung diese Datei verwendet.

  MQS_DISABLE_ALL_INTERCEPT, Umgebungsvariable
  Sie müssen diese Variable festlegen, wenn Sie IBM MQ 8.0 oder höher mit nativen Clients verwenden und AMS auf dem Client inaktivieren müssen.

  Erstellen Sie die Umgebungsvariable MQS_DISABLE_ALL_INTERCEPT und setzen Sie sie auf TRUE in der Umgebung, in der der Client ausgeführt wird. Beispiel:

  ```
  export MQS_DISABLE_ALL_INTERCEPT =TRUE
  ```

  Sie können die Umgebungsvariable MQS_DISABLE_ALL_INTERCEPT nur für C-Clients verwenden. Für Java-Clients müssen Sie stattdessen die Umgebungsvariable AMQ_DISABLE_CLIENT_AMS verwenden.

DisableClientAMS-Eigenschaft in der `mqclient.ini`-Datei

Sie können diese Option für IBM MQ classes for JMS- und IBM MQ classes for Java-Clients sowie für C-Clients verwenden.

Fügen Sie den Eigenschaftsnamen `DisableClientAMS` unter der Zeilengruppe **`Security`** die Datei `mqclient.ini` hinzu, wie im folgenden Beispiel gezeigt:

```
Security:
DisableClientAMS=Yes
```

Sie können AMS wie im folgenden Beispiel gezeigt aktivieren:

```
Security:
DisableClientAMS=No
```

Nächste Schritte

Weitere Informationen zu Problemen beim Öffnen von AMS-geschützten Warteschlangen finden Sie unter Probleme beim Öffnen von geschützten Warteschlangen bei Verwendung von AMS mit JMS.

Zugehörige Konzepte

„Abfangen des Message Channel Agent (MCA) und AMS” auf Seite 690
Durch das MCA-Abfangen kann ein Warteschlangenmanager, der unter IBM MQ ausgeführt wird, die für Serververbindungskanäle angewendeten Richtlinien gezielt aktivieren.

Zugehörige Tasks

Client mit einer Konfigurationsdatei konfigurieren

Zugehörige Verweise

Konfigurationsdatei für IBM MQ classes for JMS

Zertifikatsanforderungen für AMS

Zertifikate müssen über einen öffentlichen RSA-Schlüssel verfügen, damit sie mit Advanced Message Security verwendet werden können.

Weitere Informationen zu verschiedenen Typen öffentlicher Schlüssel und deren Erstellung finden Sie unter „Digitale Zertifikate und CipherSpec-Kompatibilität in IBM MQ” auf Seite 49.

Schlüsselverwendungserweiterungen

Die Schlüsselverwendungserweiterungen stellen zusätzliche Einschränkungen für die Verwendung eines Zertifikats dar.

In Advanced Message Security muss die Schlüsselnutzung von X.509 v3-Zertifikaten in Übereinstimmung mit der Spezifikation RFC 5280 festgelegt werden.

Für die Integritätskomponente der Datenschutzqualität müssen sie, wenn die Erweiterungen für die Verwendung von Zertifikatsschlüsseln festgelegt sind, mindestens eines der beiden Folgenden enthalten:

- **`nonRepudiation`**
- **`digitalSignature`**

Für die Geheimhaltungskomponente der Datenschutzqualität müssen sie, wenn die Erweiterungen für die Verwendung von Zertifikatsschlüsseln festgelegt sind, Folgendes enthalten:

- **`keyEncipherment`**

Für die Vertraulichkeitskomponente der Datenschutzqualität müssen sie, wenn die Erweiterungen für die Verwendung von Zertifikatsschlüsseln festgelegt sind, Folgendes enthalten:

- **`dataEncipherment`**

Durch die erweiterte Schlüsselnutzung werden die Schlüsselnutzungserweiterungen genauer definiert. Für alle Komponenten der Datenschutzqualität müssen sie, wenn die erweiterte Schlüsselnutzung für Zertifikate festgelegt ist, folgende enthalten:

- **`emailProtection`**

Zugehörige Konzepte

„Qualität des Schutzes in AMS” auf Seite 716
Advanced Message Security-Richtlinien für den Datenschutz beinhalten ein Datenschutzniveau (Quality of Protection, QOP).

Methoden zur Zertifikatsprüfung in AMS

Sie können mit Advanced Message Security widerrufene Zertifikate ermitteln und zurückweisen, damit diese Nachrichten in Ihren Warteschlangen nicht mithilfe von Zertifikaten geschützt werden, welche die Sicherheitsstandards nicht erfüllen.

Mit AMS können Sie die Zertifikatsgültigkeit mit Online Certificate Status Protocol (OCSP) oder mit der Zertifikatsperrliste (CRL) prüfen.

AMS kann für die Prüfung von OCSP und/oder der CRL konfiguriert werden. Wenn beide Methoden aktiviert werden, verwendet AMS aufgrund von Leistungsaspekten zuerst OCSP für den Widerrufsstatus. Wenn der Widerrufsstatus eines Zertifikats nach der OCSP-Prüfung nicht ermittelt werden kann, verwendet AMS die CRL-Prüfung.

Beachten Sie, dass sowohl die OCSP-als auch die CRL-Prüfung standardmäßig aktiviert sind.

Zugehörige Konzepte

„OCSP (Online Certificate Status Protocol) in AMS” auf Seite 695
OCSP (Online Certificate Status Protocol) bestimmt, ob ein Zertifikat widerrufen wurde, und hilft daher festzustellen, ob das Zertifikat anerkannt werden kann. OCSP ist standardmäßig aktiviert.

„Zertifikatswiderrufslisten (CRLs) in AMS” auf Seite 697
CRLs enthält eine Liste von Zertifikaten, die von der Zertifizierungsinstanz (CA) als nicht mehr vertrauenswürdig markiert wurden, z. B. weil der private Schlüssel verloren gegangen ist oder beeinträchtigt wurde.

OCSP (Online Certificate Status Protocol) in AMS

OCSP (Online Certificate Status Protocol) bestimmt, ob ein Zertifikat widerrufen wurde, und hilft daher festzustellen, ob das Zertifikat anerkannt werden kann. OCSP ist standardmäßig aktiviert.

OCSP wird auf IBM i-Systemen nicht unterstützt.

OCSP-Prüfung in nativen Interceptors von Advanced Message Security aktivieren
Die OCSP-Prüfung (Online Certificate Status Protocol) in Advanced Message Security wird standardmäßig auf Basis der Informationen in den verwendeten Zertifikaten aktiviert.

Vorgehensweise

Fügen Sie der Schlüsselspeicherkonfigurationsdatei die folgenden Optionen hinzu:

Anmerkung: Die gesamte OCSP-Zeilengruppe ist optional und kann unabhängig voneinander angegeben werden.

Option	Beschreibung
`ocsp.enable=off`	Aktivieren Sie die OCSP-Prüfung, wenn das Zertifikat, das überprüft wird, über eine AIA-Erweiterung (Authority Info Access) mit einer Zugriffsmethode PKIX_AD_OCSP verfügt, die eine URI enthält, in der sich der OCSP-Responder befindet. Mögliche Werte: on oder `off`.

Option	Beschreibung
`ocsp.url=`*`responder_URL`*	Die URL-Adresse des OCSP-Responder. Wenn diese Option weggelassen wird, ist die OCSP-Prüfung für Nicht-AIA inaktiviert.
`ocsp.http.proxy.host=`*`OCSP_proxy`*	Die URL-Adresse des OCSP-Proxy-Servers. Wenn diese Option weggelassen wird, wird kein Proxy für Nicht-AIA-Online-Zertifikatsprüfungen verwendet.
`ocsp.http.proxy.port=`*`port_number`*	Die Port-Nummer des OCSP-Proxy-Servers. Wenn diese Option weggelassen wird, wird der Standardport 8080 verwendet.
`ocsp.nonce.generation=on/off`	Nonce beim Abfragen von OCSP generieren. Der Standardwert ist `off`.
`ocsp.nonce.check=on/off`	Überprüfen Sie Nonce, nachdem Sie eine Antwort vom OCSP empfangen haben. Der Standardwert ist `off`.
`ocsp.nonce.size=8`	Nonce-Größe in Byte.
`ocsp.http.get=on/off`	Geben Sie HTTP GET als Anforderungsmethode an. Wenn diese Option auf `off` gesetzt ist, wird HTTP POST verwendet. Der Standardwert ist `off`.
`ocsp.max_response_size=20480`	Maximale Größe der Antwort vom OCSP-Responder in Byte.
`ocsp.cache_size=100`	Aktivieren Sie das interne OCSP-Antwort-Caching, und legen Sie den Grenzwert für die Anzahl der Cacheeinträge fest.
`ocsp.timeout=30`	Wartezeit für eine Serverantwort in Sekunden, nach der das Zeitlimit für Advanced Message Security überschritten ist.
`ocsp.unknown=ACCEPT`	Definiert das Verhalten, wenn ein OCSP-Server innerhalb eines Zeitlimitintervalls nicht erreicht werden kann. Mögliche Werte: • `ACCEPT` Ermöglicht das Zertifikat • `WARN` Ermöglicht das Zertifikat und protokolliert eine Warnung. • `REJECT` Verhindert, dass das Zertifikat verwendet wird, und protokolliert einen Fehler.

OCSP-Prüfung in Java in AMS aktivieren
Um die OCSP-Prüfung für Java in Advanced Message Security zu aktivieren, ändern Sie die Datei `java.security` oder die Konfigurationsdatei für den Schlüsselspeicher.

Informationen zu diesem Vorgang

Es gibt zwei Möglichkeiten, die OCSP-Prüfung in Advanced Message Security zu aktivieren:

Verwenden von ' java.security
Überprüfen Sie, ob Ihr Zertifikat eine AIA-Zertifikatserweiterung (Authority Information Access) enthält.

Vorgehensweise

1. Wenn AIA nicht konfiguriert ist oder Sie Ihr Zertifikat überschreiben möchten, bearbeiten Sie die Datei `$JAVA_HOME/lib/security/java.security` mit den folgenden Eigenschaften:

   ```
   ocsp.responderURL=http://url.to.responder:port
   ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
   ```

   und aktivieren Sie die OCSP-Prüfung, indem Sie die Datei `$JAVA_HOME/lib/security/java.security` mit der folgenden Zeile bearbeiten:

   ```
   ocsp.enable=true
   ```

2. Wenn AIA eingerichtet ist, aktivieren Sie die OCSP-Prüfung, indem Sie die Datei `$JAVA_HOME/lib/security/java.security` mit der folgenden Zeile bearbeiten:

   ```
   ocsp.enable=true
   ```

Nächste Schritte

Wenn Sie den Java Security Manager verwenden, um die Konfiguration abzuschließen, fügen Sie die folgende Java-Berechtigung der Datei `lib/security/java.policy` hinzu:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

Keystore.conf verwenden

Vorgehensweise

Fügen Sie das folgende Attribut zur Konfigurationsdatei hinzu:

```
ocsp.enable=true
```

Wichtig: Wenn Sie dieses Attribut in der Konfigurationsdatei festlegen, werden die Einstellungen für 'java.security' überschrieben.

Nächste Schritte

Fügen Sie zum Abschließend der Konfiguration die folgenden Java-Berechtigungen der Datei `lib/security/java.policy` hinzu:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

Zertifikatswiderrufslisten (CRLs) in AMS

CRLs enthält eine Liste von Zertifikaten, die von der Zertifizierungsinstanz (CA) als nicht mehr vertrauenswürdig markiert wurden, z. B. weil der private Schlüssel verloren gegangen ist oder beeinträchtigt wurde.

Zum Prüfen von Zertifikaten erstellt Advanced Message Security eine Zertifikatskette, die aus dem Zertifikat des Unterzeichners und der Zertifizierungskette der Zertifizierungsstelle bis zu einem Trust-Anchor besteht. Ein Trust-Anchor ist eine vertrauenswürdige Schlüsselspeicherdatei, die ein anerkantes Zertifikat oder ein Trusted-Root-Zertifikat enthält, das verwendet wird, um das Vertrauen eines Zertifikats zu bestätigen. AMS überprüft den Zertifikatspfad mit einem PKIX-Validierungsalgorithmus. Wenn die Kette erstellt und überprüft ist, schließt AMS die Zertifikatsprüfung ab. In dieser wird das Ausgabe- und Ablaufdatums jedes Zertifikats in der Kette mit dem aktuellen Datum ausgewertet und es wird überprüft, ob die Erweiterung der Schlüsselnutzung im Endentitätszertifikat vorhanden ist. Wenn die Erweiterung an das Zertifikat angehängt wird, überprüft AMS, ob auch **digitalSignature** oder **nonRepudiation** festgelegt sind. Wenn

dies nicht der Fall ist, wird der MQRC_SECURITY_ERROR gemeldet und protokolliert. Als nächstes lädt AMS die CRLs aus Dateien oder von LDAP herunter, abhängig davon, welche Werte in der Konfigurationsdatei angegeben wurde. Nur CRLs, die im DER-Format codiert sind, werden von AMS unterstützt. Wenn sich in der Konfigurationsdatei des Keystores keine CRL-bezogene Konfiguration befindet, führt AMS keine Gültigkeitsprüfung für CRLs aus. AMS fragt für jedes CA-Zertifikat den LDAP-Server nach CRLs ab und verwendet dabei die definierten Namen einer Zertifizierungsstelle, um die zugehörige CRL zu suchen. Die folgenden Attribute sind in der LDAP-Abfrage enthalten:

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
deltaRevocationList
deltaRevocationList;binary,
```

Anmerkung: `deltaRevocationList` wird nur unterstützt, wenn es als Verteilungspunkte angegeben wird.

Unterstützung für Zertifikatvalidierung und Zertifikatwiderrufsliste in nativen Interceptor aktivieren
Sie müssen die Konfigurationsdatei für den Schlüsselspeicher ändern, damit Advanced Message Security die CLRs vom LDAP-Server (Lightweight Directory Access Protocol) herunterladen kann.

Informationen zu diesem Vorgang

IBM i Die Aktivierung der Zertifikatsprüfung und der Unterstützung von Zertifikatswiderrufslisten in nativen Interceptors wird für Advanced Message Security unter IBM i nicht unterstützt.

Vorgehensweise

Fügen Sie der Konfigurationsdatei die folgenden Optionen hinzu:

Anmerkung: Die gesamte CRL-Zeilengruppe ist optional und kann unabhängig voneinander angegeben werden.

Option	Beschreibung
`crl.ldap.host=`*`host_name`*	Hostname des LDAP-Servers.
`crl.ldap.port=`*`port_number`*	Portnummer des LDAP-Servers. Sie können bis zu 11 Server angeben. Es werden mehrere LDAP-Hosts verwendet, um eine transparente Funktionsübernahme im Falle eines LDAP-Verbindungsfehlers zu gewährleisten. Es wird erwartet, dass alle LDAP-Server Replikate sind und die gleichen Daten enthalten. Wenn der AMS Java-Interceptor erfolgreich eine Verbindung zu einem LDAP-Server hergestellt hat, versucht er nicht, CRLs von den übrigen bereitgestellten Servern herunterzuladen.
`crl.cdp=off`	Verwenden Sie diese Option, um CRLDistribution-Points-Erweiterungen in Zertifikaten zu überprüfen oder zu verwenden.
`crl.ldap.version=3`	Versionsnummer des LDAP-Protokolls. Mögliche Werte: 2 oder 3.
`crl.ldap.user=cn=`*`username`*	Melden Sie sich am LDAP-Server an. Wenn dieser Wert nicht angegeben wird, müssen die CRL-Attribute in LDAP weltlesbar sein.

Option	Beschreibung
`crl.ldap.pass=`*password*	Kennwort für den LDAP-Server.
V 9.2.2 `crl.ldap.encrypted=`*no/yes*	Gibt an, ob `clr.ldap.pass` verschlüsselt ist oder nicht. Weitere Informationen finden Sie unter Kennwörter in AMS-Konfigurationsdateien schützen.
`crl.ldap.cache_lifetime=0`	Lebensdauer des LDAP-Caches in Sekunden. Mögliche Werte: 0-86400.
`crl.ldap.cache_size=50`	LDAP-Cachegröße. Diese Option kann nur angegeben werden, wenn der `crl.ldap.cache_lifetime`-Wert größer als `0`ist.
`crl.http.proxy.host=some.host.com`	HTTP-Proxy-Server-Port für CDP-CRL-Abruf.
`crl.http.proxy.port=8080`	Http-Proxy-Server-Portnummer.
`crl.http.max_response_size=204800`	Die maximale Größe der -CRL (in Byte), die von einem HTTP-Server abgerufen werden kann, der von GSKit akzeptiert wird.
`crl.http.timeout=30`	Die Wartezeit für eine Serverantwort (in Sekunden), nach der das Zeitlimit für AMS überschritten ist.
`crl.http.cache_size=0`	HTTP-Cachegröße in Byte.
`crl.unknown=ACCEPT`	Definiert das Verhalten, wenn ein CRL-Server nicht innerhalb eines Zeitlimitintervalls erreicht werden kann. Mögliche Werte: • `ACCEPT` Ermöglicht das Zertifikat • `WARN` Ermöglicht das Zertifikat und protokolliert eine Warnung. • `REJECT` Verhindert, dass das Zertifikat verwendet wird, und protokolliert einen Fehler.

Unterstützung der Zertifikatswiderrufsliste in Java in AMS aktivieren
Um die CRL-Unterstützung in Advanced Message Security zu aktivieren, müssen Sie die Keystore-Konfigurationsdatei ändern, damit AMS CRLs von dem LDAP-Server (Lightweight Directory Access Protocol) herunterladen und die Datei java.security konfigurieren kann.

Vorgehensweise

1. Fügen Sie der Konfigurationsdatei die folgenden Optionen hinzu:

Kopfzeile	Beschreibung
`crl.ldap.host=`*host_name*	LDAP-Hostname.

Kopfzeile	Beschreibung
`crl.ldap.port=`*`port_number`*	Portnummer des LDAP-Servers. Sie können bis zu 11 Server angeben. Es werden mehrere LDAP-Hosts verwendet, um eine transparente Funktionsübernahme im Falle eines LDAP-Verbindungsfehlers zu gewährleisten. Es wird erwartet, dass alle LDAP-Server Replikate sind und die gleichen Daten enthalten. Wenn der AMS Java-Interceptor erfolgreich eine Verbindung zu einem LDAP-Server hergestellt hat, versucht er nicht, CRLs von den übrigen bereitgestellten Servern herunterzuladen. Java verwendet nicht die Werte `crl.ldap.user` und crl.ldaworldp.pass. Er verwendet keinen Benutzer und kein Kennwort, wenn eine Verbindung zu einem LDAP-Server hergestellt wird. Daher müssen CRL-Attribute in LDAP weltweit lesbar sein.
`crl.cdp=on/off`	Verwenden Sie diese Option, um CRLDistribution-Points-Erweiterungen in Zertifikaten zu überprüfen oder zu verwenden.

2. Ändern Sie die JRE/`lib/security/java.security`-Datei mit den folgenden Eigenschaften:

Eigenschaftsname	Beschreibung
`com.ibm.security.enableCRLDP`	Für diese Eigenschaft werden die folgenden Werte verwendet: `true`, `false`. Wenn diese Option auf `true` gesetzt ist, werden CRLs bei der Zertifikatswiderrufsprüfung mithilfe der URL der CRL-Verteilungspunkte-Erweiterung des Zertifikats lokalisiert. Wenn die Option auf `false` gesetzt ist oder nicht festgelegt ist, ist die Überprüfung der CRL unter Verwendung der CRL-Verteilungspunkte-Erweiterung inaktiviert.
`ibm.security.certpath.ldap.cache.lifetime`	Diese Eigenschaft kann verwendet werden, um die Lebensdauer von Einträgen im Speichercache von LDAP CertStore auf einen Wert in Sekunden zu setzen. Bei einem Wert von 0 wird der Cache inaktiviert. -1 bedeutet unbegrenzte Lebensdauer. Wird diese Option nicht festgelegt, beträgt die Standardlebensdauer 30 Sekunden.

Eigenschaftsname	Beschreibung
`com.ibm.security.enableAIAEXT`	Für diese Eigenschaft werden die folgenden Werte verwendet: `true`, `false`. Wenn sie auf `true` gesetzt ist, werden alle Zugriffserweiterungen der Berechtigungsinformationen, die in den Zertifikaten des erstellten Zertifikatspfads gefunden werden, geprüft, um festzustellen, ob sie LDAP-URIs enthalten. Für jeden gefundenen LDAP-URI wird ein LDAPCertStore-Objekt erstellt und zur Sammlung von CertStores hinzugefügt, das zum Lokalisieren anderer Zertifikate verwendet wird, die zum Erstellen des Zertifikatpfads erforderlich sind. Wenn sie auf `false` gesetzt ist oder nicht festgelegt ist, werden keine zusätzlichen LDAPCertStore-Objekte erstellt.

z/OS *Zertifikatswiderrufslisten (CRLs) unter z/OS aktivieren*

Advanced Message Security unterstützt die Überprüfung der digitalen Zertifikate, mit denen Datennachrichten geschützt werden, durch eine Zertifikatswiderrufsliste (CRL).

Informationen zu diesem Vorgang

Wenn Advanced Message Security aktiviert ist, werden Empfängerzertifikate beim Einreihen von Nachrichten in eine mit 'Privacy' geschützte Warteschlange und Absenderzertifikate beim Abrufen von Nachrichten aus einer geschützten Warteschlange (Integrity oder Privacy) geprüft. In diesem Fall muss geprüft werden, ob die relevanten Zertifikate nicht in einer relevanten CRL registriert sind.

Advanced Message Security verwendet IBM System SSL-Services, um Sender- und Empfängerzertifikate zu prüfen. Eine ausführliche Dokumentation zur Prüfung von System SSL-Zertifikaten finden Sie im Handbuch z/OS Cryptographic Services System Secure Sockets Layer Programming (SC24-5901).

Um die CRL-Prüfung zu aktivieren, geben Sie die Position einer CRL-Konfigurationsdatei über das DD-Format CRLFILE in der gestarteten Task-JCL für den AMS-Adressraum an. Eine CRL-Beispielkonfigurationsdatei, die angepasst werden kann, wird in *thlqual* .SCSQPROC (CSQ40CRL) bereitgestellt. Die in dieser Datei zulässigen Einstellungen lauten wie folgt:

Tabelle 103. Konfigurationsvariablen für die Advanced Message Security-CRL

Variable	Gültige Werte	Beschreibung
crl.ldap.host [.n]	*hostname -or- hostname:port*	ipaddr/Hostname des LDAP-Servers, der CRLs Ihrer Ausstellerzertifikate hostet. Wenn Sie keine Portnummer für Ihren LDAP-Server angeben, wird die Portnummer verwendet, die von crl.ldap.port angegeben wird.
crl.ldap.port	*port*	Die TCP/IP-Anschlussnummer des LDAP-Servers.
crl.ldap.user	*ldap_user*	Der LDAP-Benutzername, der beim Herstellen der Verbindung zum LDAP-Server verwendet werden soll.

Tabelle 103. Konfigurationsvariablen für die Advanced Message Security-CRL (Forts.)		
Variable	**Gültige Werte**	**Beschreibung**
crl.ldap.pass	*ldap_password*	Das LDAP-Kennwort, das dem crl.ldap.user zugeordnet ist.

Sie können Hostnamen und Ports für mehrere LDAP-Server wie folgt angeben:

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

Sie können bis zu 10 Host-Namen angeben. Wenn Sie keine Portnummer für Ihre LDAP-Server angeben, wird die Portnummer verwendet, die von crl.ldap.port angegeben wird. Jeder LDAP-Server muss die gleiche Kombination aus crl.ldap.user/password für den Zugriff verwenden.

Wenn die DD-Datei CRLFILE angegeben ist, wird die Konfiguration während der Initialisierung des Advanced Message Security-Adressraums geladen und die CRL-Prüfung wird aktiviert. Wenn die Datendefinitionsdatei CRLFILE nicht angegeben ist oder die CRL-Konfigurationsdatei nicht verfügbar oder ungültig ist, ist die CRL-Prüfung inaktiviert.

AMS führt mithilfe der Services zur Zertifikatsprüfung von IBM System SSL eine CRL-Prüfung folgendermaßen aus:

Tabelle 104. CRL-Prüfungen von Advanced Message Security		
Operation	**Qualität des Schutzes**	**Zertifikat (en) geprüft**
EINREIHEN	Datenschutz	Empfänger (n)
GET	Integrität/Datenschutz	Sender

Wenn eine Messaging-Operation eine CRL-Prüfung nicht besteht, führt Advanced Message Security die folgenden Aktionen aus:

Tabelle 105. Verhalten beim Fehlschlagen einer CRL-Prüfung für Advanced Message Security	
Operation	**CRL-Prüffehler**
EINREIHEN	Die Nachricht wird nicht in die Zielwarteschlange gestellt. Ein Beendigungscode von MQCC_FAILED und ein Ursachencode von MQRC_SECURITY_ERROR werden an die Anwendung zurückgegeben.
GET	Die Nachricht wird aus der Zielwarteschlange entfernt und in die Fehlerwarteschlange des Systemschutzes verschoben. Ein Beendigungscode von MQCC_FAILED und ein Ursachencode von MQRC_SECURITY_ERROR werden an die Anwendung zurückgegeben.

AMS for z/OS verwendet IBM System SSL-Services zum Prüfen von Zertifikaten, die CRL-Prüfungen und Prüfungen der Vertrauenswürdigkeit umfassen. IBM System SSL stellt die Umgebungsvariable GSK_CRL_SECURITY_LEVEL bereit, um die Operation der CRL-Prüfung abzumildern. Beispiel:

```
GSK_CRL_SECURITY_LEVEL=MEDIUM
```

Diese Variable wird im Handbuch 'z/OS Cryptographic Services System Secure Sockets Layer Programming' dokumentiert. Zu den gültigen Zuordnungen gehören:

- LOW-Zertifikatprüfung schlägt nicht fehl, wenn der LDAP-Server nicht kontaktiert werden kann.

- Bei der MEDIUM-Zertifikatsprüfung ist es erforderlich, dass der LDAP-Server erreichbar ist. Es ist jedoch nicht erforderlich, dass eine CRL definiert wird.
- HIGH-Für die Validierung des Zertifikats ist es erforderlich, dass der LDAP-Server erreichbar ist und eine CRL definiert wird.

Die Standardeinstellung von IBM System SSL ist MEDIUM. Sie können diese Variable in der Konfigurationsdatei festlegen, die über das ENVARS DD in der gestarteten Task-JCL für den AMS-Adressraum angegeben ist. Eine Beispielkonfigurationsdatei für Umgebungsvariablen wird in *thlqual* .SCSQPROC (CSQ40ENV) bereitgestellt.

Anmerkung: Es liegt in der Verantwortung der Administratoren, relevante LDAP-Services bereitzustellen und die CRL-Einträge für relevante Zertifizierungsstellen zu verwalten.

V 9.2.0 V 9.2.0 AMS Kennwortschutz für -Konfigurationsdateien einrichten

Das Speichern von Kennwörtern für den Keystore und für private Schlüssel als Klartext stellt ein Sicherheitsrisiko dar, weshalb Advanced Message Security ein Tool bereitstellt, das diese Kennwörter mithilfe eines Benutzerschlüssels verschlüsselt.

Vorbereitende Schritte

Der `keystore.conf`-Dateieigner muss sicherstellen, dass nur der Dateieigner berechtigt ist, die Datei zu lesen und in die Datei zu schreiben. Der in diesem Abschnitt beschriebene Kennwortschutz ist nur eine zusätzliche Schutzmaßnahme. Darüber hinaus sollten Sie dieses Verfahren in einem sicheren System ausführen.

V 9.2.2 Stellen Sie sicher, dass Sie die richtige **`runamscred`**-Variante für den Typ des AMS-Clients verwenden, der die Konfigurationsdatei lesen soll. Handelt es sich bei dem AMS-Client um:

- Java -Client sollten Sie den Befehl Java **`runamscred`** verwenden, der sich im Verzeichnis `<IBM MQ installation root>/java/bin` befindet.
- MQI-Client, Sie sollten den MQI- **`runmqascred`** -Befehl verwenden, der sich im Verzeichnis `<IBM MQ installation root>/bin` befindet

Vorgehensweise

1. Bearbeiten Sie die `keystore.conf`-Dateien so, dass sie alle erforderlichen Informationen, einschließlich der Kennwörter, die geschützt werden müssen, einschließen.

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

2. Platzieren Sie den Verschlüsselungsschlüssel zur Verschlüsselung der Passwörter in einer Datei, auf die der Benutzer, der die `keystore.conf` Datei schützt, Zugang hat.

   V 9.2.2 Dieser Schlüssel muss der gleiche Schlüssel sein, der später vom AMS-Client verwendet werden soll:

```
ThisIsAnExampleEncryptionKey
```

3. Führen Sie den Befehl **`runamscred`** aus, um die Datei `keystore.conf` zu schützen, in der der Verschlüsselungsschlüssel bereitgestellt wird.

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. Stellen Sie sicher, dass die `keystore.conf`-Datei geschützt wurde und verschlüsselte Kennwörter enthält.

Beispiel

Das folgende Beispiel zeigt, wie eine geschützte `keystore.conf`-Datei aussieht:

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = yes
jceks.keystore_pass =
<AMS>1!62K/a4RinT+bks4RjFWx4A==!Vhi/RjIN2FH5qStUJ/0hsgKyn2IdMuhanemRRDrJq
HM=
jceks.key_pass =
<AMS>1!qmnxY++rsOUtZfDSgwcR1g==!VmWVREdVkNp1xYJstvuW64ph5vxxf7SPoqtsXxYh2
Tk=
jceks.provider = IBMJCE
```

Zugehörige Informationen

runamscred: AMS-Schlüsselwörter schützen

z/OS Zertifikate mit AMS unter z/OS verwenden

Informationen zu diesem Vorgang

Advanced Message Security implementiert drei Schutzebenen: Integrität, Vertraulichkeit und Datenschutz.

Bei einer Integritätsrichtlinie werden Nachrichten mit dem privaten Schlüssel des Erstellers signiert (die Anwendung, die den MQPUT-Befehl ausgeführt wird). Integrität ermöglicht die Erkennung von Nachrichtenänderungen, aber der Nachrichtentext selbst ist nicht verschlüsselt.

Bei einer Vertraulichkeitsrichtlinie wird die Nachricht verschlüsselt, wenn sie in die Warteschlange gestellt wird. Die Nachricht wird mit einem symmetrischen Schlüssel und einem Algorithmus verschlüsselt, der in der zugehörigen Advanced Message Security-Richtlinie angegeben ist. Der symmetrische Schlüssel selbst wird mit dem öffentlichen Schlüssel jedes Empfängers verschlüsselt (die Anwendung, die den MQGET-Aufruf ausgeführt hat). Öffentliche Schlüssel werden mit Zertifikaten verknüpft, die in Schlüsselringen gespeichert sind.

Bei einer Datenschutzrichtlinie werden Nachrichten signiert und verschlüsselt.

Wenn eine Nachricht, die geschützt ist, durch eine Empfängeranwendung, die einen MQGET-Aufruf ausgeführt hat, in die Warteschlange gestellt wird, muss die Nachricht entschlüsselt werden. Da die Verschlüsselung mit dem öffentlichen Schlüssel des Empfängers verschlüsselt wurde, muss sie mit dem privaten Schlüssel des Empfängers entschlüsselt werden, der in einem Schlüsselring gefunden wurde.

z/OS *Verwendung von SAF-Schlüsselringen mit AMS unter z/OS*

Advanced Message Security (AMS) nutzt z/OS-SAF-Schlüsselringservices zum Definieren und Verwalten der Zertifikate, die für die Signierung und Verschlüsselung erforderlich sind. Sicherheitsprodukte, die funktional äquivalent zu RACF sind, können anstelle von RACF verwendet werden, wenn sie die gleiche Unterstützungsstufe bereitstellen.

Die effiziente Verwendung von Schlüsselringen kann die Verwaltung reduzieren, die für die Verwaltung der Zertifikate benötigt wird.

Nachdem ein Zertifikat generiert (oder importiert) wurde, muss es mit einem Schlüsselring verbunden werden, um zugänglich zu werden. Das gleiche Zertifikat kann an mehrere Schlüsselringe angeschlossen werden.

Advanced Message Security verwendet zwei Gruppen von Schlüsselringen. Eine Gruppe setzt sich aus Schlüsselringen zusammen, deren Eigner die einzelnen Benutzer-IDs sind, die Nachrichten senden oder empfangen. Jeder Schlüsselring enthält den privaten Schlüssel, der dem Zertifikat der Eignerbenutzer-ID zugeordnet ist. Der private Schlüssel jedes Zertifikats wird zum Signieren von Nachrichten für Integritätsschutz-oder Datenschutz-geschützte Warteschlangen verwendet. Es wird auch zum Entschlüsseln von Nachrichten aus geschützten oder vertraulichen Datenschutzwarteschlangen verwendet, die beim Empfang von Nachrichten geschützt sind.

Bei der anderen Gruppe handelt es sich um einen einzelnen Schlüsselring, dessen Eigner der AMS -Adressraumbenutzer ist. Sie enthält die Kette der Signieren von CA-Zertifikaten, die zum Prüfen der Zertifikate des Nachrichtenturstellers und der Empfänger erforderlich sind.

Wenn der Datenschutz oder der Vertraulichkeitsschutz verwendet wird, enthält der Schlüsselring, dessen Eigner der AMS-Adressraumbenutzers ist, auch die Zertifikate der Nachrichtenempfänger. Die öffentlichen Schlüssel in diesen Zertifikaten werden verwendet, um den symmetrischen Schlüssel zu verschlüsseln, der zum Verschlüsseln der Nachrichtendaten verwendet wurde, als die Nachricht in die geschützte Warteschlange gestellt wurde. Wenn diese Nachrichten abgerufen werden, wird der private Schlüssel der relevanten Empfänger verwendet, um den symmetrischen Schlüssel zu entschlüsseln, der dann zur Entschlüsselung der Nachrichtendaten verwendet wird.

Advanced Message Security verwendet den Schlüsselringnamen **drq.ams.keyring** bei der Suche nach Zertifikaten und privaten Schlüsseln. Dies ist sowohl für den Benutzer als auch für die AMS-Adressraumschlüsselringe der Fall.

Eine Abbildung und weitere Erläuterungen zu Zertifikaten und Schlüsselanschlüssen sowie ihre Rolle im Datenschutz finden Sie im Abschnitt Zusammenfassung der zertifikatsbezogenen Operationen .

Der private Schlüssel, der zum Signieren und Entschlüsseln verwendet wird, kann einen beliebigen Kennsatz aufweisen, muss aber als Standardzertifikat verbunden sein.

Digitale Zertifikate und Schlüsselringe werden in RACF in erster Linie mit dem Befehl RACDCERT verwaltet.

Weitere Informationen zu Zertifikaten, Bezeichnungen und dem Befehl RACDCERT finden Sie in den Handbüchern *z/OS: Security Server RACF Command Language Reference* und *z/OS: Security Server RACF Security Administrator's Guide*.

z/OS *Zugriff auf den RACDCERT-Befehl für AMS unter z/OS autorisieren*

Die Berechtigung für die Verwendung des Befehls RACDCERT ist eine Task, die nach der Installation ausgeführt wird und die von Ihrem z/OS-Systemprogrammierer ausgeführt werden sollte. In dieser Task werden dem Advanced Message Security-Sicherheitsadministrator die relevanten Berechtigungen erteilt.

Zusammengefasst sind diese Befehle erforderlich, um den Zugriff auf den RACF-Befehl RACDCERT zu ermöglichen:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID( admin ) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

In diesem Beispiel gibt *admin* die Benutzer-ID Ihres Sicherheitsadministrators oder einen beliebigen Benutzer an, den Sie mit dem Befehl RACDCERT verwenden möchten.

z/OS *Zertifikate und Schlüsselringe für AMS-Benutzer unter z/OS erstellen*

In diesem Abschnitt werden die Schritte dokumentiert, die zum Erstellen der Zertifikate und Schlüsselringe erforderlich sind, die für z/OS-Benutzer von Advanced Message Security (AMS) benötigt werden, die eine RACFZertifizierungsstelle (CA) verwenden.

Probleme mit Zertifikaten bei der Verwendung von Advanced Message Security unter z/OS beheben

Wenn Sie Probleme mit Zertifikaten und fehlenden Einträgen in Schlüsselspeichern haben, können Sie einen GSKIT-Trace aktivieren.

Fügen Sie in der Datei, auf die von der ENVARS DD in der Prozedur für gestartete Tasks von AMS verwiesen wird, Folgendes hinzu:

```
GSK_TRACE_FILE=/u/... /gsktrace
GSK_TRACE=0xff
```

Weitere Informationen finden Sie unter Umgebungsvariablen.

Für jeden Zugriff auf den Keystore werden Daten in die Tracedatei geschrieben, die in GSK_TRA-CE_FILEangegeben ist.

Verwenden Sie den folgenden Befehl, um die Tracedatei zu formatieren

```
gsktrace inputtrace file > output_file
```

Szenario

Ein Szenario mit einer sendenden Anwendung und einer empfangenden Anwendung wird verwendet, um die erforderlichen Schritte zu erläutern.

In den folgenden Beispielen ist `user1` der Ersteller einer Nachricht und `user2` der Empfänger. Die Benutzer-ID des Advanced Message Security-Adressraums ist `WMQAMSD`.

Alle Befehle in den hier gezeigten Beispielen werden von der ISPF-Option 6 durch die ID des Benutzers mit Verwaltungsaufgaben `admin` ausgegeben.

z/OS *Zertifikat der lokalen Zertifizierungsinstanz für AMS unter z/OS definieren*

Wenn Sie RACF als Zertifizierungsstelle verwenden, müssen Sie ein Zertifikat einer Zertifizierungsstelle erstellen, wenn Sie dies noch nicht getan haben. Mit dem hier angezeigten Befehl wird ein Zertifikat für die Zertifizierungsstelle (oder Unterzeichner) erstellt. In diesem Beispiel wird ein Zertifikat mit der Bezeichnung AMSCA erstellt, das beim Erstellen von nachfolgenden Zertifikaten verwendet werden soll, die die Identität von Advanced Message Security-Benutzern und -Anwendungen darstellen.

Dieser Befehl kann geändert werden, insbesondere SUBJECTSDN, um die Namensstruktur und die bei der Installation verwendeten Konventionen zu berücksichtigen:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('AMSCA') O('ibm') C('us'))
KEYUSAGE(CERTSIGN) WITHLABEL('AMSCA')
```

Anmerkung: Zertifikate, die mit diesem Zertifikat der lokalen Zertifizierungsstelle signiert sind, zeigen einen Aussteller von CN=AMSCA, O=ibm, C=us an, wenn dieser mit dem Befehl RACDCERT LIST aufgelistet wird.

z/OS *Erstellen eines digitalen Zertifikats mit einem privaten Schlüssel für AMS unter z/OS*

Für jeden Advanced Message Security-Benutzer muss ein digitales Zertifikat mit einem privaten Schlüssel erstellt werden. In dem hier gezeigten Beispiel werden RACDCERT-Befehle verwendet, um Zertifikate für Benutzer 1 und Benutzer2 zu generieren, die mit dem lokalen CA-Zertifikat signiert werden, das durch die Bezeichnung AMSCA identifiziert wird.

```
RACDCERT ID(user1) GENCERT SUBJECTSDN(CN('user1') O('ibm') C('us'))
WITHLABEL('user1') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user2) GENCERT SUBJECTSDN(CN('user2') O('ibm') C('us'))
WITHLABEL('user2') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user1) ALTER (LABEL('user1')) TRUST
RACDCERT ID(user2) ALTER (LABEL('user2')) TRUST
```

Der Befehl RACDCERT ALTER ist erforderlich, um das Attribut TRUST zu dem Zertifikat hinzuzufügen. Wenn ein Zertifikat zuerst mit dieser Prozedur erstellt wird, hat es einen anderen gültigen Datumsbereich als das Signaturzertifikat. Daher wird es von RACF als NOTRUST markiert, was bedeutet, dass das Zertifikat nicht verwendet werden soll. Verwenden Sie den Befehl RACDCERT ALTER, um das Attribut TRUST festzulegen.

Die KEYUSAGE-Attribute HANDSHAKE, DATAENCRYPT und DOCSIGN müssen für Zertifikate angegeben werden, die von Advanced Message Security verwendet werden.

Tabelle 106. Werte und Anzeiger für RACDCERT KEYUSAGE

KEYUSAGE-Wert	**Bezugszahlen**
HANDSHAKE	digitalSignature und keyEncipherment
DATAENCRYPT	dataEncipherment
DOCSIGN	nonRepudiation
CERTSIGN	keyCertSign und cRLSign

z/OS *RACF -Schlüsselringe für AMS unter z/OS erstellen*

Mit den hier gezeigten Befehlen wird ein Schlüsselring für die mit RACF definierten Benutzer-IDs 'user1', 'user2' und den Benutzer WMQAMSD der Tasks für den Advanced Message Security-Adressraum erstellt. Der Name des Schlüsselrings wird von Advanced Message Security festgelegt und muss wie dargestellt ohne Anführungszeichen codiert werden. Bei dem Namen muss die Groß-/Kleinschreibung beachtet werden.

```
RACDCERT ID(user1) ADDRING(drq.ams.keyring)
RACDCERT ID(user2) ADDRING(drq.ams.keyring)
RACDCERT ID(WMQAMSD) ADDRING(drq.ams.keyring)
```

z/OS *Zertifikate mit den Schlüsselringen für AMS unter z/OS verbinden*

Verbinden Sie die Benutzer-und CA-Zertifikate mit den Schlüsselringen:

```
RACDCERT ID(WMQAMSD) CONNECT(CERTAUTH LABEL('AMSCA')
RING(drq.ams.keyring))
RACDCERT ID(user1) CONNECT(ID(user1) LABEL('user1')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(user2) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(WMQAMSD) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) USAGE(SITE))
```

Das Zertifikat, das den privaten Schlüssel enthält, der für die Entschlüsselung verwendet wird, muss mit dem Schlüsselring des Benutzers als Standardzertifikat verbunden sein.

Das Attribut RACDCERT USAGE (SITE) verhindert, dass der private Schlüssel im Schlüsselring zugänglich ist, während das Attribut RACDCERT USAGE (PERSONAL) zulässt, dass der private Schlüssel verwendet wird, wenn er vorhanden ist. Das Benutzer2-Zertifikat muss mit dem Advanced Message Security-Adressraum-Schlüsselring verbunden sein, da sein öffentlicher Schlüssel zum Verschlüsseln von Nachrichten, die in die Warteschlange eingereiht werden, erforderlich ist. USAGE (SITE) begrenzt die Exposition des privaten Schlüssels von user2.

Das CERTAUTH-Zertifikat mit der Bezeichnung AMSCA muss mit dem Schlüsselring des Advanced Message Security-Adressraums verbunden sein, da es zum Signieren des Zertifikats von 'user1' verwendet wurde, bei dem es sich um den Absender der Nachricht handelt. Es wird verwendet, um das Signaturzertifikat von user1 zu validieren.

z/OS *Schlüsselringprüfung für AMS unter z/OS*

Der Schlüsselring sollte wie hier dargestellt angezeigt werden, nachdem alle Befehle eingegeben wurden:

```
RACDCERT ID(user1) LISTRING(drq.ams.keyring)
Digital ring information for user USER1:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user1                          ID(USER1)    PERSONAL YES

RACDCERT ID(user2) LISTRING(drq.ams.keyring)
Digital ring information for user USER2:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user2                          ID(USER2)    PERSONAL YES

RACDCERT ID(WMQAMSD) LISTRING(drq.ams.keyring)
Digital ring information for user WMQAMSD:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
AMSCA                          CERTAUTH     CERTAUTH NO
user2                          ID(USER2)    SITE     NO
```

Die Zuordnung der einzelnen Zertifikate zeigt auch die Ringzuordnung.

```
RACDCERT ID(user2) LIST(label('user2'))
Digital certificate information for user USER2:

***
Label: user2
Certificate ID: 2QfH8Pny9/LzpKKFmfFA
Status: TRUST
Start Date: 2010/05/03 22:59:53
End Date:  2011/05/04 22:59:52
Serial Number:>15<:
Issuer's Name:>OU=AMSCA.O=ibm.C=us<:
Subject's Name:>CN=user2.O=ibm.C=us<:
Key Usage: HANDSHAKE, DATAENCRYPT, DOCSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
Ring Owner: USER2
Ring:>drq.ams.keyring<:
Ring Owner: WMQAMSD
Ring:>drq.ams.keyring<:
```

Um die Leistung zu verbessern, wird der Inhalt des drq.ams.keyring, der dem AMS-Adressraum zugeordnet ist, für die Lebensdauer des Adressraums zwischengespeichert. Änderungen an diesem Schlüsselring werden nicht automatisch wirksam. Der Administrator kann den Cache wie folgt aktualisieren:

- Der WS-Manager wird gestoppt und erneut gestartet.
- z/OS-Befehl MODIFY verwenden:

```
F qmgrAMSM,REFRESH KEYRING
```

Zugehörige Tasks

Advanced Message Security ausführen

z/OS *Zusammenfassung der zertifikatsbezogenen Operationen für AMS unter z/OS*

Abbildung 35 auf Seite 709 veranschaulicht die Beziehungen zwischen dem Senden und Empfangen von Anwendungen und relevanten Zertifikaten. Das dargestellte Szenario umfasst die ferne Steuerung der

Warteschlangensteuerung zwischen zwei z/OS-Warteschlangenmanagern unter Verwendung einer Datenschutzrichtlinie für den Datenschutz. In Abbildung 35 auf Seite 709 gibt "AMS" den Wert "Advanced Message Security" an.

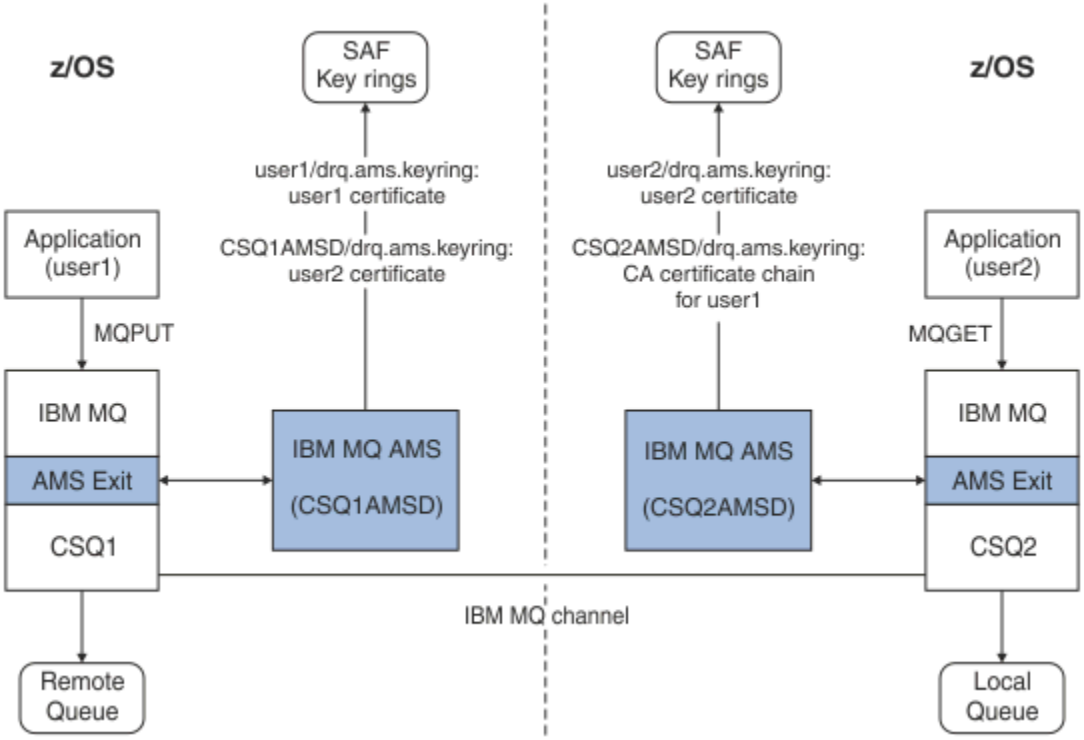


Abbildung 35. Anwendungs-und Zertifikatbeziehungen

In diesem Diagramm stellt eine Anwendung, die als 'user1' ausgeführt wird, eine Nachricht in eine ferne Warteschlange ein, die vom Warteschlangenmanager CSQ1 verwaltet wird, der von einer Anwendung abgerufen werden soll, die als 'user2' aus einer lokalen Warteschlange ausgeführt wird, die vom WS-Manager CSQ2 verwaltet wird. Das Diagramm setzt eine Advanced Message Security-Datenschutzrichtlinie voraus. Dies bedeutet, dass die Nachricht signiert und verschlüsselt ist.

Advanced Message Security fängt die Nachricht bei einer Einreihung ab und verwendet das Zertifikat von Benutzer 'user2' (im Schlüsselring des AMS-Adressraums gespeichert), um einen symmetrischen Schlüssel zu verschlüsseln, mit dem die Nachrichtendaten verschlüsselt werden.

Beachten Sie, dass das Benutzerzertifikat von 'user2' mit dem Benutzerschlüsselring des AMS-Adressraums mit der Option USAGE (SITE) verbunden ist. Dies bedeutet, dass der AMS-Adressraum-Benutzer auf das Zertifikat und den öffentlichen Schlüssel zugreifen kann, aber nicht den privaten Schlüssel.

Auf der Empfängerseite fängt Advanced Message Security die von 'user2' ausgegebene GET-Anforderung ab und entschlüsselt mit dem Zertifikat von 'user2' den symmetrischen Schlüssel, damit die Nachrichtendaten entschlüsselt werden können. Anschließend wird die Signatur von user1 anhand der CA-Zertifikatskette des Benutzerzertifikats überprüft, die im Schlüsselring des AMS-Adressraumbenutzers gespeichert ist.

In diesem Szenario, aber mit einer Datenschutzrichtlinie für die Integrität, sind die Zertifikate für Benutzer2 nicht erforderlich.

Wenn Sie Advanced Message Security verwenden möchten, um Nachrichten in eine von IBM MQ geschützte Warteschlange einzureihen, die über die Nachrichtenschutzrichtlinie 'Privacy' oder 'Integrity' verfügt, muss Advanced Message Security auf die folgenden Datenelemente zugreifen können:

- Das X.509 V2-oder V3-Zertifikat und der private Schlüssel für den Benutzer, der die Nachricht in die Warteschlange eingereiht hat.

- Die Kette der Zertifikate, die zum Signieren der digitalen Zertifikate aller Nachrichtensignatoren verwendet werden.
- Handelt es sich bei der Datenschutzrichtlinie um Datenschutz, handelt es sich bei dem X.509 V2-oder V3-Zertifikat der vorgesehenen Empfänger um eine Datenschutzrichtlinie. Die vorgesehenen Empfänger werden in der Advanced Message Security-Richtlinie aufgeführt, die der Warteschlange zugeordnet ist.

Für Prozesse und Anwendungen, die auf z/OS ausgeführt werden, muss Advanced Message Security über Zertifikate an zwei Positionen verfügen:

- In einem mit SAF verwalteten Schlüsselring, der der RACF-Identität der sendenden Anwendung (mit der die geschützte Nachricht in die Warteschlange gestellt wird) oder der empfangenden Anwendung (bei der Verwendung von 'Privacy') zugeordnet ist.

  Das Zertifikat, das von Advanced Message Security gesucht wird, ist das Standardzertifikat und muss den privaten Schlüssel enthalten. Advanced Message Security setzt voraus, dass die z/OS-Benutzeridentität der sendenden Anwendung vorhanden ist. Das heißt, es handelt sich um ein Ersatzzeichen, so dass es auf den privaten Schlüssel des Benutzers zugreifen kann.
- In einem SAF-verwalteten Schlüsselring, der dem AMS-Adressraumbenutzer zugeordnet ist.

  Beim Senden von Nachrichten, die mit Datenschutz geschützt sind, enthält dieser Schlüsselring die öffentlichen Schlüsselzertifikate der Nachrichtenempfänger. Beim Empfang von Nachrichten enthält es die Kette der Zertifikate der Zertifizierungsstelle, die zum Prüfen der Signatur des Nachrichtensenders erforderlich sind.

In den vorherigen Beispielen wurde RACF als lokale Zertifizierungsstelle verwendet. Sie können jedoch bei Ihrer Installation einen anderen PKI-Provider (Certificate Authority) verwenden. Wenn Sie ein anderes PKI-Produkt verwenden möchten, denken Sie daran, dass der private Schlüssel und das Zertifikat in einen Schlüsselring importiert werden müssen, der den z/OS RACF -Benutzer-IDs zugeordnet ist, die von IBM MQ Nachrichten stammen, die durch Advanced Message Securitygeschützt werden.

Sie können den RACF-Befehl RACDCERT als Mechanismus zum Generieren von Zertifikatsanforderungen verwenden, die exportiert und zur Ausgabe an den PKI-Provider Ihrer Wahl gesendet werden können.

Im Folgenden finden Sie eine Zusammenfassung der zertifikatbezogenen Schritte:

1. Fordern Sie die Erstellung eines CA-Zertifikats an, in dem RACF die lokale Zertifizierungsstelle ist. Übergehen Sie diesen Schritt, wenn Sie einen anderen PKI-Provider verwenden.
2. Generieren Sie die von der Zertifizierungsstelle signierten Benutzerzertifikate.
3. Erstellen Sie die Schlüsselringe für die Benutzer und die Adressraum-ID für Advanced Message Security AMS.
4. Verbinden Sie das Benutzerzertifikat mit dem Benutzerschlüsselring mit dem Standardattribut.
5. Verbinden Sie die Empfängerzertifikate mithilfe des Attributs 'usage(site)' mit dem Benutzerschlüsselring des Advanced Message Security AMS-Adressraums. (Dieser Schritt ist nur für Benutzerzertifikate erforderlich, bei denen es sich letztlich um die Empfänger von Nachrichten mit dem Schutz 'Privacy' handelt).
6. Verbinden Sie die CA-Zertifikatskette für Nachrichtensender mit dem Benutzerschlüssel des Advanced Message Security AMS-Adressraums. (Dieser Schritt ist nur für AMS-Tasks erforderlich, die die Sendersignaturen prüfen.)

z/OS *Nicht z/OS-residente PKI für AMS konfigurieren*

Advanced Message Security for z/OS verwendet digitale Zertifikate des Typs X.509 V3 bei der geschützten Verarbeitung von Nachrichten, die in IBM MQ-Warteschlangen eingereiht oder von dort abgerufen werden. Advanced Message Security erstellt oder verwaltet den Lebenszyklus dieser Zertifikate nicht selbst; diese Funktion wird von einer PKI-Infrastruktur bereitgestellt. Die Beispiele in dieser Veröffentlichung, mit denen die Verwendung von Zertifikaten dargestellt werden, verwenden z/OS Security Server RACF, um Zertifikatsanforderungen auszufüllen.

Unabhängig davon, ob sich die PKI Infrastruktur auf einem z/OS-System oder einem anderen System als z/OS befindet, verwendet AMS for z/OS nur Schlüsselringe, die von RACF oder einer Entsprechung verwal-

tet werden. Diese Schlüsselringe basieren auf Security Authorization Facility (SAF) und es handelt sich dabei um das Repository, mit dem AMS for z/OS Zertifikate für Absender und Empfänger von Nachrichten erhält, die in IBM MQ-Warteschlangen gestellt oder von dort empfangen werden.

Für Nachrichten aus z/OS, die mit der Richtlinie 'Integrity' oder 'Encryption' verschlüsselt sind, müssen das Zertifikat und der private Schlüssel der Benutzer-ID des Erstellers in einem von SAF verwalteten Schlüsselring gespeichert werden, der der z/OS-Benutzer-ID des Nachrichtenabsenders zugeordnet ist.

RACF umfasst die Funktion zum Importieren von Zertifikaten und privaten Schlüsseln in Schlüsselringe, die von RACF verwaltet werden. In den Veröffentlichungen zu z/OS Security Server RACF finden Sie Informationen und Beispiele zum Laden von Zertifikaten in den von RACF verwalteten Schlüsselringen.

Wenn Ihre Installation eines der unterstützten PKI-Produkte verwendet, lesen Sie die Veröffentlichungen, die mit dem Produkt verbunden sind, um Informationen darüber zu erhalten, wie Sie die Produkte implementieren.

Advanced Message Security-Sicherheitsrichtlinien anwenden

Advanced Message Security verwendet Sicherheitsrichtlinien, um die Verschlüsselungs- und Signaturalgorithmen für die Verschlüsselung und Authentifizierung von Nachrichten anzugeben, die durch die Warteschlangen fließen.

Übersicht über die Sicherheitsrichtlinien für AMS

Bei Advanced Message Security-Sicherheitsrichtlinien handelt es sich um konzeptionelle Objekte, mit denen beschrieben wird, wie eine Nachricht verschlüsselt und signiert wird.

Ausführliche Informationen zu den Attributen der Sicherheitsrichtlinie finden Sie in den folgenden Unterabschnitten:

Zugehörige Konzepte

„Qualität des Schutzes in AMS” auf Seite 716
Advanced Message Security-Richtlinien für den Datenschutz beinhalten ein Datenschutzniveau (Quality of Protection, QOP).

„Sicherheitsrichtlinienattribute in AMS” auf Seite 715
Sie können mit Advanced Message Security einen bestimmten Algorithmus auswählen, mit dem die Daten geschützt werden.

Richtliniennamen in AMS

Der Richtlinienname ist ein eindeutiger Name, der eine bestimmte Advanced Message Security-Richtlinie und die Warteschlange angibt, auf die sie angewendet wird.

Der Richtlinienname muss mit dem Namen der Warteschlange übereinstimmen, für die er gilt. Es gibt eine Eins-zu-eins-Zuordnung zwischen einer Advanced Message Security-Richtlinie (AMS) und einer Warteschlange.

Wenn Sie eine Richtlinie mit demselben Namen wie eine Warteschlange erstellen, aktivieren Sie die Richtlinie für diese Warteschlange. Warteschlangen ohne übereinstimmende Richtliniennamen werden nicht durch AMS geschützt.

Der Geltungsbereich der Richtlinie ist für den lokalen WS-Manager und dessen Warteschlangen relevant. Ferne Warteschlangenmanager müssen über eigene, lokal definierte Richtlinien für die Warteschlangen verfügen, die sie verwalten.

Signaturalgorithmus in AMS

Der Signaturalgorithmus gibt den Algorithmus an, der beim Signieren von Datennachrichten verwendet werden soll.

Folgende Werte sind gültig:

- MD5
- SHA-1

- SHA-2 -Produktfamilie:
  - SHA256
  - SHA384 (Mindestschlüssellänge akzeptabel-768 Bit)
  - SHA512 (Mindestschlüssellänge akzeptabel-768 Bit)

Eine Richtlinie, die keinen Signaturalgorithmus angibt oder einen Algorithmus von NONE angibt, impliziert, dass Nachrichten, die in die Warteschlange gestellt werden, die der Richtlinie zugeordnet ist, nicht signiert sind.

Anmerkung: Die Qualität des Schutzes, der für die Nachrichten-und Absendungsfunktionen verwendet wird, muss übereinstimmen. Wenn eine Richtlinienqualität der Zugriffsschutzabweichung zwischen der Warteschlange und der Nachricht in der Warteschlange vorhanden ist, wird die Nachricht nicht akzeptiert und an die Fehlerbehandlungswarteschlange gesendet. Diese Regel gilt sowohl für lokale als auch für ferne Warteschlangen.

Verschlüsselungsalgorithmus in AMS

Der Verschlüsselungsalgorithmus zeigt den Algorithmus an, der beim Verschlüsseln von Datennachrichten verwendet werden soll, die in die Warteschlange gestellt werden, die der Richtlinie zugeordnet ist.

Folgende Werte sind gültig:

- RC2
- DES
- 3DES
- AES128
- AES256

Eine Richtlinie, die keinen Verschlüsselungsalgorithmus angibt oder einen Algorithmus von NONE angibt, impliziert, dass die Nachrichten, die in die der Richtlinie zugeordnete Warteschlange gestellt werden, nicht verschlüsselt sind.

Beachten Sie, dass eine Richtlinie, die einen anderen Verschlüsselungsalgorithmus als NONE angibt, außerdem mindestens einen Empfänger-DN und einen Signaturalgorithmus angeben muss, da auch verschlüsselte Advanced Message Security-Nachrichten signiert werden.

Wichtig: Die Qualität des Schutzes, der für die Nachrichten-und Absendungsfunktionen verwendet wird, muss übereinstimmen. Wenn eine Richtlinienqualität der Zugriffsschutzabweichung zwischen der Warteschlange und der Nachricht in der Warteschlange vorhanden ist, wird die Nachricht nicht akzeptiert und an die Fehlerbehandlungswarteschlange gesendet. Diese Regel gilt sowohl für lokale als auch für ferne Warteschlangen.

Toleranz in AMS

Das Toleranzattribut zeigt an, ob Advanced Message Security Nachrichten akzeptieren kann, für die keine Sicherheitsrichtlinie angegeben ist.

Wenn eine Nachricht aus einer Warteschlange mit einer Richtlinie zum Verschlüsseln von Nachrichten abgerufen wird, wird sie an die aufrufenden Anwendung zurückgegeben, wenn die Nachricht nicht verschlüsselt ist. Folgende Werte sind gültig:

0
Nein (**Standardwert**).

1
Ja.

Eine Richtlinie, die keinen Toleranzwert angibt oder 0 angibt, impliziert, dass Nachrichten, die in die Warteschlange gestellt werden, die der Richtlinie zugeordnet ist, mit den Richtlinienregeln übereinstimmen müssen.

Die Toleranz ist optional und ist vorhanden, um das Rollout der Konfiguration zu vereinfachen, wobei Richtlinien auf Warteschlangen angewendet wurden, diese Warteschlangen jedoch bereits Nachrichten enthalten, für die keine Sicherheitsrichtlinie angegeben ist.

Definierte Namen des Senders in AMS

Die definierten Namen (DNs) des Absenders identifizieren Benutzer, die berechtigt sind, Nachrichten in eine Warteschlange einzureihen. Ein Absender verwendet sein Zertifikat zum Signieren einer Nachricht, bevor die Nachricht in eine Warteschlange eingereiht wird.

Advanced Message Security (AMS) prüft nicht, ob eine Nachricht von einem gültigen Benutzer in eine datengeschützte Warteschlange eingereiht wurde, bis die Nachricht abgerufen wird. Wenn die Richtlinie mindestens einen gültigen Absender festlegt und der Benutzer, der die Nachricht in die Warteschlange gestellt hat, nicht in der Liste der gültigen Absender enthalten ist, gibt AMS zu diesem Zeitpunkt einen Fehler an die empfangende Anwendung zurück und stellt die Nachricht in die AMS-Fehlerwarteschlange.

Eine Richtlinie kann 0 oder mehr Absender-DNs haben. Wenn keine Absender-DNs für die Richtlinie angegeben sind, kann jeder Absender datengeschützte Nachrichten in die Warteschlange stellen, sofern das Zertifikat des Absenders vertrauenswürdig ist. Das Zertifikat eines Absenders wird anerkannt, indem das öffentliche Zertifikat einem Keystore hinzugefügt wird, der der empfangenden Anwendung zur Verfügung steht.

Absenderdefinierte Namen haben das folgende Format:

```
CN=Common Name,O=Organization,C=Country
```

Wichtig:

- Alle DN-Komponentennamen müssen in Großbuchstaben angegeben werden. Alle Komponentennamenskennungen im DN müssen in der in der folgenden Tabelle angegebenen Reihenfolge angegeben werden:

Komponentenname	Wert
CN	Der allgemeine Name für das Objekt dieses DN, wie z. B. ein vollständiger Name oder der beabsichtigte Zweck einer Einheit.
OU	Die Einheit innerhalb der Organisation, mit der das Objekt des definierten Namens verbunden ist, z. B. eine Unternehmensdivision oder ein Produktname.
O	Die Organisation, mit der das Objekt des registrierten Namens verbunden ist, z. B. eine Firma.
L	Die Lokalität (Stadt oder Gemeinde), in der sich das Objekt des DN befindet.
ST	Der Name des Landes oder der Provinz, in dem sich das Objekt des DN befindet.
C	Das Land, in dem sich das Objekt des registrierten Namens (DN) befindet.

- Wenn ein oder mehrere Absender-DNs für die Richtlinie angegeben sind, können nur diese Benutzer Nachrichten in die Warteschlange einlegen, die der Richtlinie zugeordnet ist.
- Absender-DNs müssen, wenn angegeben, exakt mit dem DN übereinstimmen, der in dem digitalen Zertifikat enthalten ist, das dem Benutzer zugeordnet ist, der die Nachricht eingibt.
- AMS unterstützt nur Werte für definierte Namen aus dem Zeichensatz 'Lateinisches Alphabet 1'. Wenn Sie DNs mit Zeichen dieses Zeichensatzes erstellen möchten, müssen Sie zuerst ein Zertifikat mit einem DN erstellen, der in UTF-8-Codierung unter Verwendung von AIX and Linux mit aktiver UTF-8-Codierung oder mit der **strmqikm**-GUI erstellt wird. Anschließend müssen Sie eine Richtlinie über eine Linux-

oder eine AIX-Plattform mit aktivierter UTF-8-Codierung erstellen oder das AMS-Plug-in für IBM MQ verwenden.

- Die von AMS verwendete Methode zum Konvertieren des Sendernamens vom x.509-Format in das Format des definierten Namens verwendet immer ST= für den Wert des Staates oder des Bundeslands.
- Die folgenden Sonderzeichen müssen Escapezeichen enthalten:

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- Wenn der definierte Name eingebettete Leerzeichen enthält, müssen Sie den DN in doppelte Anführungszeichen setzen.

Zugehörige Konzepte

„Definierte Namen des Empfängers in AMS" auf Seite 714
Die definierten Namen (DN) des Empfängers geben Benutzer an, die berechtigt sind, Nachrichten aus einer Warteschlange abzurufen.

Definierte Namen des Empfängers in AMS

Die definierten Namen (DN) des Empfängers geben Benutzer an, die berechtigt sind, Nachrichten aus einer Warteschlange abzurufen.

Eine Richtlinie kann null oder mehr Empfänger-DNs angeben. Empfängerdefinierte Namen haben das folgende Format:

```
CN=Common Name,O=Organization,C=Country
```

Wichtig:

- Alle DN-Komponentennamen müssen in Großbuchstaben angegeben werden. Alle Komponentennamenskennungen im DN müssen in der in der folgenden Tabelle angegebenen Reihenfolge angegeben werden:

Komponentenname	Wert
CN	Der allgemeine Name für das Objekt dieses DN, wie z. B. ein vollständiger Name oder der beabsichtigte Zweck einer Einheit.
OU	Die Einheit innerhalb der Organisation, mit der das Objekt des definierten Namens verbunden ist, z. B. eine Unternehmensdivision oder ein Produktname.
O	Die Organisation, mit der das Objekt des registrierten Namens verbunden ist, z. B. eine Firma.
L	Die Lokalität (Stadt oder Gemeinde), in der sich das Objekt des DN befindet.
ST	Der Name des Landes oder der Provinz, in dem sich das Objekt des DN befindet.
C	Das Land, in dem sich das Objekt des registrierten Namens (DN) befindet.

- Wenn keine Empfänger-DNs für die Richtlinie angegeben sind, kann jeder Benutzer Nachrichten aus der Warteschlange abrufen, die der Richtlinie zugeordnet ist.

- Wenn ein oder mehrere Empfänger-DNs für die Richtlinie angegeben sind, können nur die Benutzer Nachrichten aus der Warteschlange abrufen, die der Richtlinie zugeordnet ist.
- Empfänger-DNs müssen, wenn sie angegeben werden, genau dem DN entsprechen, der in dem digitalen Zertifikat enthalten ist, das dem Benutzer zugeordnet ist, der die Nachricht erhält.
- Advanced Message Security unterstützt nur Werte für definierte Namen aus dem Zeichensatz 'Lateinisches Alphabet 1'. Um DNs mit Zeichen aus dem Set zu erstellen, müssen Sie zunächst ein Zertifikat mit DN mit UTF-8-Codierung unter AIX oder Linux mit aktivierter UTF-8-Codierung oder mit der **strmqikm**-GUI erstellen. Anschließend müssen Sie eine Richtlinie über eine Linux- oder eine AIX-Plattform mit aktivierter UTF-8-Codierung erstellen oder das Advanced Message Security-Plug-in für IBM MQ verwenden.

Zugehörige Konzepte

„Definierte Namen des Senders in AMS" auf Seite 713
Die definierten Namen (DNs) des Absenders identifizieren Benutzer, die berechtigt sind, Nachrichten in eine Warteschlange einzureihen. Ein Absender verwendet sein Zertifikat zum Signieren einer Nachricht, bevor die Nachricht in eine Warteschlange eingereiht wird.

Sicherheitsrichtlinienattribute in AMS

Sie können mit Advanced Message Security einen bestimmten Algorithmus auswählen, mit dem die Daten geschützt werden.

Eine Sicherheitsrichtlinie ist ein konzeptionelles Objekt, das beschreibt, wie eine Nachricht verschlüsselt verschlüsselt und signiert wird.

Tabelle 107. Sicherheitsrichtlinienattribute in AMS

Attribute	**Beschreibung**
Richtlinienname	Eindeutiger Name der Richtlinie für einen WS-Manager.
Signaturalgorithmus	Verschlüsselungsalgorithmus, der zum Signieren von Nachrichten vor dem Senden verwendet wird.
Verschlüsselungsalgorithmus	Verschlüsselungsalgorithmus, der zum Verschlüsseln von Nachrichten vor dem Senden verwendet wird.
Empfängerliste (Recipient)	Liste der registrierten Namen (DNs) von Zertifikaten für potenzielle Empfänger einer Nachricht.
Prüfliste für Signatur-DN	Liste der Signatur-DNs, die während des Nachrichtenabrufs geprüft werden sollen.

In Advanced Message Security werden Nachrichten mit einem symmetrischen Schlüssel verschlüsselt und der symmetrische Schlüssel ist mit den öffentlichen Schlüsseln des Empfängers verschlüsselt. Öffentliche Schlüssel werden mit dem RSA-Algorithmus verschlüsselt, wobei die Schlüssel eine effektive Länge von bis zu 2048 Bits haben. Die tatsächliche asymmetrische Schlüsselchiffrierung hängt von der Länge des Zertifikatschlüssels ab.

Es werden folgende symmetrische Schlüsselalgorithmen unterstützt:

- RC2
- DES
- 3DES
- AES128
- AES256

Advanced Message Security unterstützt auch die folgenden kryptografischen Hashfunktionen:

- MD5

- SHA-1
- SHA-2 -Produktfamilie:
  - SHA256
  - SHA384 (Mindestschlüssellänge akzeptabel-768 Bit)
  - SHA512 (Mindestschlüssellänge akzeptabel-768 Bit)

Anmerkung: Die Qualität des Schutzes, der für die Nachrichten-und Absendungsfunktionen verwendet wird, muss übereinstimmen. Wenn eine Richtlinienqualität der Zugriffsschutzabweichung zwischen der Warteschlange und der Nachricht in der Warteschlange vorhanden ist, wird die Nachricht nicht akzeptiert und an die Fehlerbehandlungswarteschlange gesendet. Diese Regel gilt sowohl für lokale als auch für ferne Warteschlangen.

Qualität des Schutzes in AMS

Advanced Message Security-Richtlinien für den Datenschutz beinhalten ein Datenschutzniveau (Quality of Protection, QOP).

Die drei Datenschutzniveau-Ebenen in Advanced Message Security werden in IBM MQ 9.0 und höher um eine vierte Ebene ergänzt und hängen von Verschlüsselungsalgorithmen ab, die zum Signieren und Verschlüsseln der Nachricht verwendet werden:

- Datenschutz-Nachrichten, die in die Warteschlange gestellt werden, müssen signiert und verschlüsselt werden.
- Integrity-Nachrichten, die in die Warteschlange gestellt werden, müssen vom Absender signiert werden.
- Vertraulichkeitsnachrichten-Nachrichten, die in die Warteschlange gestellt werden, müssen verschlüsselt werden. Weitere Informationen hierzu finden Sie in „Für AMS verfügbare Datenschutzniveaus” auf Seite 636.
- Kein-kein Datenschutz ist anwendbar.

Eine Richtlinie, die festlegt, dass Nachrichten signiert werden müssen, wenn sie in eine Warteschlange gestellt werden, verfügt über einen QOP von INTEGRITY. Ein QOP von INTEGRITY bedeutet, dass eine Richtlinie einen Signaturalgorithmus festlegt, aber keinen Verschlüsselungsalgorithmus festlegt. Integrity-geschützte Nachrichten werden auch als "SIGNED" bezeichnet.

Eine Richtlinie, die festlegt, dass Nachrichten signiert und verschlüsselt werden müssen, wenn sie in eine Warteschlange gestellt werden, verfügt über einen QOP von PRIVACY. Ein QOP von PRIVACY bedeutet, dass bei einer Richtlinie ein Signaturalgorithmus und ein Verschlüsselungsalgorithmus festgelegt werden. Vertraulichkeitsgeschützte Nachrichten werden auch als "SEALED" bezeichnet.

Eine Richtlinie, die festlegt, dass Nachrichten verschlüsselt werden müssen, wenn sie in eine Warteschlange gestellt werden, die über einen QOP von CONFIDENTIALITY verfügt. Ein QOP von CONFIDENTIALITY bedeutet, dass eine Richtlinie einen Verschlüsselungsalgorithmus festlegt.

Eine Richtlinie, die keinen Signaturalgorithmus oder einen Verschlüsselungsalgorithmus festlegt, weist einen QOP von NONE auf. Advanced Message Security stellt keinen Datenschutz für Warteschlangen bereit, die über eine QOP mit dem Wert NONE verfügen.

Sicherheitsrichtlinien in AMS verwalten

Eine Sicherheitsrichtlinie ist ein konzeptionelles Objekt, das beschreibt, wie eine Nachricht verschlüsselt verschlüsselt und signiert wird.

Die Position, aus der alle Verwaltungstasks im Zusammenhang mit Sicherheitsrichtlinien ausgeführt werden, hängt von der verwendeten Plattform ab.

- ALW Verwenden Sie unter AIX, Linux, and Windows die Befehle DELETE POLICY, DISPLAY POLICY und SET POLICY (oder äquivalente PCF-Befehle), um Ihre Sicherheitsrichtlinien zu verwalten.

  - Linux AIX Unter AIX and Linux können Verwaltungstasks aus *MQ_INSTALLATION_PATH*/bin ausgeführt werden.
  - Windows Auf Windows-Plattformen können Verwaltungstasks aus jeder Position ausgeführt werden, da die Umgebungsvariable PATH bei der Installation aktualisiert wird.
- IBM i Unter IBM i sind die Befehle DSPMQMSPL, SETMQMSPL und WRKMQMSPL in der QSYS-Systembibliothek für die Primärsprache des Systems installiert, wenn IBM MQ installiert ist.

  Zusätzliche landessprachliche Versionen werden entsprechend der Sprachenladefunktion in QSYS29xx-Bibliotheken installiert. Beispiel: Eine Maschine mit amerikanischem Englisch als Primärsprache und Koreanisch als Sekundärsprache verfügt über die in QSYS installierten amerikanischen englischen Befehle und die koreanische Sekundärsprachenlast in QSYS2962 als 2962 ist die Sprache, die für Koreanisch geladen wird.
- z/OS Unter z/OS werden die Verwaltungsbefehle mit der Nachrichtensicherheitsrichtlinie (CSQ0UTIL) ausgeführt. Wenn Richtlinien unter z/OS erstellt, geändert oder gelöscht werden, werden die Änderungen erst von Advanced Message Security erkannt, wenn der Warteschlangenmanager gestoppt und erneut gestartet wird oder wenn der z/OS-Befehl MODIFY zum Aktualisieren der Advanced Message Security-Richtlinienkonfiguration verwendet wird. Beispiel:

```
F <qmgr ssid>AMSM,REFRESH POLICY
```

Zugehörige Tasks

„Sicherheitsrichtlinien in AMS erstellen” auf Seite 717
Sicherheitsrichtlinien definieren die Art und Weise, in der eine Nachricht geschützt wird, wenn die Nachricht ausgegeben wird, oder wie eine Nachricht geschützt worden sein muss, wenn eine Nachricht empfangen wird.

„Sicherheitsrichtlinien in AMS ändern” auf Seite 718
Mit Advanced Message Security können Sie Einzelheiten der Sicherheitsrichtlinien ändern, die Sie bereits definiert haben.

„Sicherheitsrichtlinien in AMS anzeigen und löschen” auf Seite 719
Mit dem Befehl **dspmqspl** können Sie eine Liste aller Sicherheitsrichtlinien oder Einzelheiten zu einer benannten Richtlinie in Abhängigkeit von den bereitgestellten Befehlszeilenparametern anzeigen.

„Sicherheitsrichtlinien in AMS entfernen” auf Seite 721
Wenn Sie Sicherheitsrichtlinien in Advanced Message Security entfernen möchten, müssen Sie den Befehl `setmqspl` verwenden.

Advanced Message Security ausführen

Zugehörige Verweise

Das Dienstprogramm für die Nachrichtensicherheitsrichtlinie (CSQ0UTIL)

Sicherheitsrichtlinien in AMS erstellen

Sicherheitsrichtlinien definieren die Art und Weise, in der eine Nachricht geschützt wird, wenn die Nachricht ausgegeben wird, oder wie eine Nachricht geschützt worden sein muss, wenn eine Nachricht empfangen wird.

Vorbereitende Schritte

Es gibt einige Eingangsbedingungen, die beim Erstellen von Sicherheitsrichtlinien erfüllt werden müssen:

- Der WS-Manager muss aktiv sein.
- Der Name einer Sicherheitsrichtlinie muss den Regeln für die Benennung von IBM MQ-Objekten entsprechen.
- Sie müssen über die erforderliche Berechtigung verfügen, um eine Verbindung zum WS-Manager herzustellen und eine Sicherheitsrichtlinie zu erstellen:

– z/OS Erteilen Sie unter z/OS die Berechtigungen, die im Abschnitt Das Dienstprogramm für die Nachrichtensicherheitsrichtlinie (CSQ0UTIL) dokumentiert sind.

– Multi Auf anderen Plattformen als z/OS müssen Sie die erforderlichen Berechtigungen +connect, +inq und +chg mit dem Befehl setmqaut erteilen.

Weitere Informationen zum Konfigurieren der Sicherheit finden Sie unter „Sicherheit konfigurieren” auf Seite 141.

- z/OS Stellen Sie unter z/OS sicher, dass die erforderlichen Systemobjekte gemäß der Definitionen in CSQ4INSM definiert wurden.

Beispiel

Im Folgenden finden Sie ein Beispiel für die Erstellung einer Richtlinie für den Warteschlangenmanager QMGR. Die Richtlinie gibt an, dass Nachrichten mit dem SHA256 -Algorithmus signiert und mit dem AES256 -Algorithmus für Zertifikate mit dem DN CN=joe, O = IBM, C=US und DN verschlüsselt werden: CN=jane, O = IBM, C = US. Diese Richtlinie ist MY.QUEUE zugeordnet:

```
setmqspl -m QMGR -p MY.QUEUE -s SHA256 -e AES256 -r CN=joe,O=IBM,C=US -r CN=jane,O=IBM,C=US
```

Im Folgenden finden Sie ein Beispiel für die Erstellung einer Richtlinie auf dem Warteschlangenmanager QMGR. Die Richtlinie gibt an, dass Nachrichten mit dem 3DES-Algorithmus für Zertifikate mit den definierten Namen CN=john,O=IBM,C=US und CN=jeff,O=IBM,C=US verschlüsselt und mit dem SHA256-Algorithmus für Zertifikate mit dem definierten Namen CN=phil,O=IBM,C=US signiert sind.

```
setmqspl -m QMGR -p MY.OTHER.QUEUE -s SHA256 -e 3DES -r CN=john,O=IBM,C=US -r
CN=jeff,O=IBM,C=US -a CN=phil,O=IBM,C=US
```

Anmerkung:

- Die Qualität des Schutzes, der für die Nachrichteneinteilung und -besicherung verwendet wird, muss übereinstimmen. Wenn die Richtlinienqualität des Schutzes, die für die Nachricht definiert ist, schwächer ist als für eine Warteschlange definiert, wird die Nachricht an die Fehlerbehandlungswarteschlange gesendet. Diese Richtlinie ist sowohl für lokale als auch für ferne Warteschlangen gültig.

Zugehörige Verweise

Vollständige Liste der setmqspl-Befehlsattribute

Sicherheitsrichtlinien in AMS ändern

Mit Advanced Message Security können Sie Einzelheiten der Sicherheitsrichtlinien ändern, die Sie bereits definiert haben.

Vorbereitende Schritte

- Der Warteschlangenmanager, auf dem Sie den Betrieb ausführen möchten, muss aktiv sein.
- Sie müssen über die erforderliche Berechtigung zum Herstellen einer Verbindung zum WS-Manager und zum Erstellen einer Sicherheitsrichtlinie verfügen.

  – z/OS Erteilen Sie unter z/OS die Berechtigungen, die im Abschnitt Das Dienstprogramm für die Nachrichtensicherheitsrichtlinie (CSQ0UTIL) dokumentiert sind.

  – Multi Auf anderen Plattformen als z/OS müssen Sie die erforderlichen Berechtigungen +connect, +inq und +chg mit dem Befehl setmqaut erteilen.

  Weitere Informationen zum Konfigurieren der Sicherheit finden Sie unter „Sicherheit konfigurieren” auf Seite 141.

Informationen zu diesem Vorgang

Zum Ändern von Sicherheitsrichtlinien wenden Sie den Befehl `setmqspl` für eine bereits vorhandene Richtlinie an und stellen neue Attribute bereit.

Beispiel

Im folgenden Beispiel wird eine Richtlinie mit dem Namen MYQUEUE auf einem Warteschlangenmanager mit dem Namen QMGRerstellt und angegeben, dass Nachrichten mit dem 3DES -Algorithmus für Autoren (**-a**) verschlüsselt werden sollen, die Zertifikate mit dem definierten Namen (DN) CN=alice, O=IBM, C=US haben und mit dem SHA256 -Algorithmus für Empfänger (**-r**) signiert sind, die Zertifikate mit dem DN CN=jeff, O=IBM, C = US haben.

```
setmqspl -m QMGR -p MYQUEUE -e 3DES -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

Um diese Richtlinie zu ändern, geben Sie den Befehl `setmqspl` mit allen Attributen aus dem Beispiel aus, die nur die Werte ändern, die Sie ändern möchten. In diesem Beispiel wird eine zuvor erstellte Richtlinie an eine neue Warteschlange angehängt und ihr Verschlüsselungsalgorithmus wird in AES256 geändert:

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

Zugehörige Verweise

setmqspl (Sicherheitsrichtlinie festlegen)

Sicherheitsrichtlinien in AMS anzeigen und löschen

Mit dem Befehl **dspmqspl** können Sie eine Liste aller Sicherheitsrichtlinien oder Einzelheiten zu einer benannten Richtlinie in Abhängigkeit von den bereitgestellten Befehlszeilenparametern anzeigen.

Vorbereitende Schritte

- Für die Anzeige der Einzelheiten der Sicherheitsrichtlinien muss der Warteschlangenmanager vorhanden und aktiv sein.
- Sie müssen über die erforderliche Berechtigung zum Herstellen einer Verbindung zum WS-Manager und zum Erstellen einer Sicherheitsrichtlinie verfügen.
  - z/OS Erteilen Sie unter z/OS die Berechtigungen, die im Abschnitt Das Dienstprogramm für die Nachrichtensicherheitsrichtlinie (CSQ0UTIL) dokumentiert sind.
  - Multi Auf anderen Plattformen als z/OS müssen Sie die erforderlichen Berechtigungen +connect, +inq und +chg mit dem Befehl setmqaut erteilen.

  Weitere Informationen zum Konfigurieren der Sicherheit finden Sie unter „Sicherheit konfigurieren” auf Seite 141.

Informationen zu diesem Vorgang

Die folgende Liste enthält die **dspmqspl** -Befehlsflags:

*Tabelle 108. **dspmqspl** -Befehlsflags.*

Befehlsmarkierung	Beschreibung
-m	Name des Warteschlangenmanagers (obligatorisch).
-p	Richtlinienname.
-export	Durch das Hinzufügen dieses Flags wird eine Ausgabe generiert, die problemlos auf einen anderen WS-Manager angewendet werden kann.

Beispiel

Im folgenden Beispiel wird gezeigt, wie zwei Sicherheitsrichtlinien für `venus.queue.manager` erstellt werden:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s sha256 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s sha256 -a "CN=another signer,O=IBM,C=US"
-e NONE
```

Dieses Beispiel zeigt einen Befehl, in dem Details zu allen für `venus.queue.manager` definierten Richtlinien und der von ihm ausgegebenen Richtlinien angezeigt werden:

```
dspmqspl -m venus.queue.manager

Policy Details:
Policy name: AMS_POL_04_ONE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=signer1,O=IBM,C=US
Recipient DNs: -
Toleration: 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

Das folgende Beispiel zeigt einen Befehl, der Details zu einer ausgewählten Sicherheitsrichtlinie anzeigt, die für `venus.queue.manager` definiert ist, und die Ausgabe, die sie erzeugt:

```
dspmqspl -m venus.queue.manager -p AMS_POL_06_THREE

Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

Im nächsten Beispiel wird zuerst eine Sicherheitsrichtlinie erstellt, und anschließend wird die Richtlinie mit dem Flag **`-export`** exportiert:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s SHA256 -a "CN=signer1,O=IBM,C=US" -e NONE

dspmqspl -m venus.queue.manager -export
```

z/OS Unter z/OS werden die Informationen zur exportierten Richtlinie von CSQ0UTIL in EXPORT DD geschrieben.

Multi Auf anderen Plattformen als z/OS leiten Sie die Ausgabe in eine Datei um, z. B.:

```
dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

So importieren Sie eine Sicherheitsrichtlinie:

- Linux AIX Unter AIX and Linux:

1. Melden Sie sich als Benutzer an, der zur Verwaltungsgruppe von `mqm` IBM MQ gehört.
2. Setzen Sie `. policies.sh` ab.

- Windows Führen Sie unter Windows den Befehl `policies.bat` aus.
- z/OS Verwenden Sie unter z/OS das Dienstprogramm CSQ0UTIL, um das Dataset für SYSIN anzugeben, das die Informationen zur exportierten Richtlinie enthält.

Zugehörige Verweise
Vollständige Liste der Attribute des Befehls 'dspmqspl'

Sicherheitsrichtlinien in AMS entfernen
Wenn Sie Sicherheitsrichtlinien in Advanced Message Security entfernen möchten, müssen Sie den Befehl `setmqspl` verwenden.

Vorbereitende Schritte
Es gibt einige Eingangsbedingungen, die beim Verwalten von Sicherheitsrichtlinien erfüllt werden müssen:

- Der WS-Manager muss aktiv sein.
- Sie müssen über die erforderliche Berechtigung zum Herstellen einer Verbindung zum WS-Manager und zum Erstellen einer Sicherheitsrichtlinie verfügen.
  - z/OS Erteilen Sie unter z/OS die Berechtigungen, die im Abschnitt Das Dienstprogramm für die Nachrichtensicherheitsrichtlinie (CSQ0UTIL) dokumentiert sind.
  - Multi Auf anderen Plattformen als z/OS müssen Sie die erforderlichen Berechtigungen +connect, +inq und +chg mit dem Befehl setmqaut erteilen.

  Weitere Informationen zum Konfigurieren der Sicherheit finden Sie unter „Sicherheit konfigurieren” auf Seite 141.

Informationen zu diesem Vorgang
Verwenden Sie den Befehl **`setmqspl`** mit der Option **`-remove`** .

Beispiel

Im Folgenden ist ein Beispiel für das Entfernen einer Richtlinie enthalten:

```
setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

Zugehörige Verweise
Vollständige Liste der setmqspl-Befehlsattribute

Schutz der Systemwarteschlange in AMS

Systemwarteschlangen aktivieren die Kommunikation zwischen IBM MQ und den zugehörigen Nebenanwendungen. Wenn ein Warteschlangenmanager erstellt wird, wird auch immer eine Systemwarteschlange erstellt, in der interne IBM MQ-Nachrichten und -Daten gespeichert werden. Sie können Systemwarteschlangen mit Advanced Message Security schützen, damit nur berechtigte Benutzer darauf zugreifen oder die Informationen entschlüsseln können.

Der Schutz der Systemwarteschlange folgt dem gleichen Muster wie der Schutz von regulären Warteschlangen. Siehe „Sicherheitsrichtlinien in AMS erstellen” auf Seite 717.

Windows Wenn Sie den Systemwarteschlangenschutz für Windowsverwenden möchten, kopieren Sie die `keystore.conf`-Datei in das folgende Verzeichnis:

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS Um unter z/OSSchutz für SYSTEM.ADMIN.COMMAND.QUEUEbereitzustellen, muss der Befehlsserver Zugriff auf die keystore und die keystore.conffhaben, die Schlüssel und eine Konfiguration enthalten, damit der Befehlsserver auf Schlüssel und Zertifikate zugreifen kann. Alle Änderungen, die an der Sicherheitsrichtlinie von SYSTEM.ADMIN.COMMAND.QUEUE vorgenommen wurden, erfordern den Neustart des Befehlsservers.

Alle Nachrichten, die von der Befehlswarteschlange gesendet und empfangen werden, werden abhängig von den Richtlinieneinstellungen signiert oder signiert und verschlüsselt. Wenn ein Administrator berechtigte Unterzeichner definiert, werden Befehlsnachrichten, die die Überprüfung der definieren Namen (DN) des Unterzeichners nicht bestehen, nicht vom Befehlsserver ausgeführt und nicht an die Advanced Message Security-Warteschlange zur Fehlerbehandlung weitergeleitet. Nachrichten, die als Antwort auf temporäre dynamische Warteschlange des IBM MQ Explorer gesendet werden, werden nicht von AMS geschützt.

Sicherheitsrichtlinien wirken sich nicht auf die folgenden SYSTEM-Warteschlangen aus:

- SYSTEM.ADMIN.ACCOUNTING.QUEUE
- SYSTEM.ADMIN.ACTIVITY.QUEUE
- SYSTEM.ADMIN.CHANNEL.EVENT
- SYSTEM.ADMIN.COMMAND.EVENT
- z/OS SYSTEM.ADMIN.COMMAND.QUEUE
- SYSTEM.ADMIN.CONFIG.EVENT
- SYSTEM.ADMIN.LOGGER.EVENT
- SYSTEM.ADMIN.PERFM.EVENT
- SYSTEM.ADMIN.PUBSUB.EVENT
- SYSTEM.ADMIN.QMGR.EVENT
- SYSTEM.ADMIN.STATISTICS.QUEUE
- SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- SYSTEM.AUTH.DATA.QUEUE
- SYSTEM.BROKER.ADMIN.STREAM
- z/OS SYSTEM.BROKER.CLIENTS.DATA
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- z/OS SYSTEM.BROKER.SUBSCRIPTIONS.DATA
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- z/OS SYSTEM.CHLAUTH.DATA.QUEUE
- SYSTEM.CICS.INITIATION.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE
- z/OS SYSTEM.COMMAND.INPUT
- z/OS SYSTEM.DDELAY.LOCAL.QUEUE
- SYSTEM.DEAD.LETTER.QUEUE

- SYSTEM.DURABLE.SUBSCRIBER.QUEUE
- SYSTEM.HIERARCHY.STATE
- SYSTEM.INTER.QMGR.CONTROL
- SYSTEM.INTER.QMGR.FANREQ
- SYSTEM.INTER.QMGR.PUBS
- SYSTEM.INTERNAL.REPLY.QUEUE
- z/OS SYSTEM.JMS.PS.STATUS.QUEUE
- z/OS SYSTEM.JMS.REPORT.QUEUE
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- z/OS SYSTEM.QSG.CHANNEL.SYNCQ
- z/OS SYSTEM.QSG.TRANSMIT.QUEUE
- z/OS SYSTEM.QSG.UR.RESOLUTION.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE
- z/OS SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

Multi V 9.2.3 Streaming-Warteschlangen und AMS

Es ist möglich, duplizierte geschützte Nachrichten aus Advanced Message Security (AMS) im Datenstrom zu übertragen.

Wenn für eine Warteschlange eine AMS-Richtlinie definiert wurde, die besagt, dass Nachrichten in dieser Warteschlange signiert und/oder verschlüsselt sein müssen, können Sie auch das Attribut **STREAMQ** der Warteschlange so konfigurieren, dass zu jeder geschützten Nachricht eine Kopie in eine zweite Warteschlange gestellt wird. Die duplizierte gestreamte Nachricht wird anhand derselben Richtlinie signiert und/oder verschlüsselt, die für die erste Warteschlange konfiguriert wurde.

Im folgenden Beispiel werden zwei Warteschlangen konfiguriert: QUEUE1 und QUEUE2. Das Attribut **STREAMQ** von QUEUE1 wird so konfiguriert, dass gestreamte Nachrichten zu QUEUE2 hinzugefügt werden:

```
DEFINE QLOCAL(QUEUE2)

DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

Geschützte Nachrichten aus AMS werden von einem Benutzer mit dem Zertifikat `CN=bob,O=IBM,C=GB` in QUEUE1 eingereiht.

Eine Anwendung mit dem Zertifikat `CN=alice,O=IBM,C=GB` verarbeitet die Nachrichten aus QUEUE1. Eine andere Anwendung mit dem Zertifikat `CN=fred,O=IBM,C=GB` verarbeitet die Nachrichten aus QUEUE2.

Auf QUEUE1 wird die folgende AMS-Datenschutzrichtlinie angewendet:

```
SET POLICY(QUEUE1) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=alice,O=IBM,C=GB') RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD
```

Wenn in der Richtlinie für QUEUE1 ein Verschlüsselungsalgorithmus konfiguriert wurde, müssen die in der Richtlinie angegebenen Empfänger sowohl die Empfänger der Originalnachrichten aus QUEUE1 als auch die Empfänger umfassen, welche die duplizierten Nachrichten aus QUEUE2 verarbeiten.

Wenn die Anwendung versucht, Nachrichten aus QUEUE2 zu verarbeiten, führt sie Integritätsprüfungen durch und/oder entschlüsselt die Nachrichten auf der Grundlage der Richtlinie, die für QUEUE2 konfiguriert wurde. Wenn eine Anwendung gestreamte Nachrichten aus QUEUE2 verarbeiten möchte, müssen Sie eine geeignete Richtlinie für QUEUE2 konfigurieren, auf deren Grundlage die Nachrichten auf ihre Integrität geprüft und ordnungsgemäß entschlüsselt werden können.

Dabei müssen insbesondere der Signieralgorithmus, der Unterzeichner und der Verschlüsselungsalgorithmus der Richtlinie entsprechen, die auf QUEUE1 angewendet wird. Die Empfänger in der Richtlinie für QUEUE2 müssen die Identität des Empfängers angeben, der Nachrichten aus QUEUE2 verarbeitet.

Anmerkung: Es ist nicht erforderlich, dass die auf QUEUE2 angewendete Richtlinie alle Empfänger enthält, die in der Richtlinie für QUEUE1 angegeben sind.

Beispielsweise könnte die folgende Richtlinie für QUEUE2 konfiguriert sein, damit die Anwendung mit dem Zertifikat `CN=fred,O=IBM,C=GB` AMS-geschützte Nachrichten aus dieser Warteschlange lesen kann:

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

Zugehörige Konzepte

Streaming-Warteschlangen

OAM-Berechtigungen in AMS gewähren

Dateiberechtigungen berechtigen alle Benutzer, `setmqspl` -und `dspmqspl` -Befehle auszuführen. Advanced Message Security basiert jedoch auf dem Objektberechtigungsmanager (OAM), und jeder Versuch, diese Befehle von einem Benutzer auszuführen, der nicht zur Gruppe 'mqm' (IBM MQ-Verwaltungsgruppe) gehört oder über keine Berechtigung zum Lesen von Sicherheitsrichtlinieneinstellungen verfügt, führt zu einem Fehler.

Vorgehensweise

Führen Sie die folgenden Schritte aus, um einem Benutzer die erforderlichen Berechtigungen zu erteilen:

```
setmqaut -m SOME.QUEUE.MANAGER -t qmgr -p SOME.USER +connect +inq
setmqaut -m SOME.QUEUE.MANAGER -t queue -n SYSTEM.PROTECTION.POLICY.QUEUE -p SOME.USER +browse
+put
setmqaut -m SOME.QUEUE.MANAGER -t queue -n SYSTEM.PROTECTION.ERROR.QUEUE -p SOME.USER +put
```

Anmerkung: Sie müssen nur diese OAM-Berechtigungen festlegen, wenn Sie Clients mithilfe von Advanced Message Security 7.0.1 mit dem Warteschlangenmanager verbinden möchten.

Achtung: Die Berechtigung zum Durchsuchen in der Warteschlange SYSTEM.PROTECTION.POLICY.QUEUE ist nicht in allen Situationen obligatorisch. IBM MQ optimiert die Leistung durch das Caching von Richtlinien, sodass Sie keine Datensätze nach Richtliniendetails auf dem SYSTEM.PROTECTION.POLICY.QUEUE in allen Fällen.

Von IBM MQ werden nicht alle verfügbaren Richtlinien zwischengespeichert. Wenn eine hohe Anzahl an Richtlinien vorhanden ist, wird von IBM MQ eine begrenzte Anzahl von Richtlinien zwischengespeichert. Wenn also der Warteschlangenmanager eine geringe Anzahl von Richtlinien definiert hat, ist es nicht erforderlich, die Option zum Durchsuchen für SYSTEM.PROTECTION.POLICY.QUEUE bereitzustellen.

Sie sollten jedoch die Berechtigung zum Durchsuchen dieser Warteschlange erteilen, falls eine hohe Anzahl an Richtlinien definiert ist, oder wenn Sie alte Clients verwenden. Die Warteschlange SYSTEM.PROTECTION.ERROR.QUEUE wird zum Einreihen von Fehlernachrichten verwendet, die vom AMS-Code generiert werden. Die Einreihungsberechtigung für diese Warteschlange wird nur überprüft, wenn Sie versuchen, eine Fehlernachricht in die Warteschlange einzureihen. Ihre Einreihungsberechtigung für die Warteschlange wird nicht überprüft, wenn Sie versuchen, eine Nachricht in eine AMS-geschützte Warteschlange einzureihen oder daraus abzurufen.

Sicherheitsberechtigungen in AMS erteilen

Bei der Verwendung der Sicherheit für die Befehlsressourcen müssen Sie Berechtigungen einrichten, damit Advanced Message Security ausgeführt werden kann. In diesem Abschnitt werden die RACF-Befehle in den Beispielen verwendet. Wenn Ihr Unternehmen einen anderen externen Sicherheitsmanager (ESM) verwendet, müssen Sie die entsprechenden Befehle für diesen ESM verwenden.

Es gibt drei Aspekte für die Erteilung von Sicherheitsberechtigungen:

- „Der AMSM-Adressraum” auf Seite 725
- „CSQ0UTIL” auf Seite 725
- „Warteschlangen verwenden, für die eine Advanced Message Security-Richtlinie definiert ist” auf Seite 726

Anmerkungen: Die Beispielbefehle verwenden die folgenden Variablen.

1. *QMgrName* -Der Name des Warteschlangenmanagers.

   z/OS Unter z/OS kann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.
2. *username* -Dies kann ein Gruppenname sein.
3. Die Beispiele zeigen die Klasse MQQUEUE. Dies kann auch MXQUEUE, GMQQUEUE oder GMXQUEUE sein. Weitere Informationen finden Sie in „Profile für die Warteschlangensicherheit” auf Seite 214 .

Wenn das Profil bereits vorhanden ist, benötigen Sie außerdem den Befehl RDEFINE nicht.

Der AMSM-Adressraum

Sie müssen eine gewisse IBM MQ-Sicherheit für den Benutzername ausgeben, unter dem der Advanced Message Security-Adressraum ausgeführt wird.

- Geben Sie für die Stapelverbindung zum Warteschlangenmanager ein Problem an.

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
           PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- Geben Sie für den Zugriff auf die Warteschlange SYSTEM.PROTECTION.POLICY.QUEUEein:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

CSQ0UTIL

Das Dienstprogramm, das es Benutzern ermöglicht, die Befehle **setmqspl** und **dspmqspl** auszuführen, erfordert die folgenden Berechtigungen, wobei der Benutzername die Jobbenutzer-ID ist:

- Geben Sie für die Stapelverbindung zum Warteschlangenmanager Folgendes ein:

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
           PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- Für den Zugriff auf die Warteschlange SYSTEM.PROTECTION.POLICY.QUEUE, die für den Befehl **setmqpol** erforderlich ist, geben Sie Folgendes aus:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
            PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- Für den Zugriff auf die Warteschlange SYSTEM.PROTECTION.POLICY.QUEUE, die für den Befehl **dspmqpol** erforderlich ist, geben Sie Folgendes aus:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
                  PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

Warteschlangen verwenden, für die eine Advanced Message Security-Richtlinie definiert ist

Wenn eine Anwendung mit Warteschlangen arbeitet, für die eine Richtlinie definiert ist, sind für die Anwendung zusätzliche Berechtigungen erforderlich, damit Advanced Message Security Nachrichten schützen kann.

Für die Anwendung ist Folgendes erforderlich:

- Lesezugriff auf SYSTEM.PROTECTION.POLICY.QUEUE. Führen Sie dazu die folgenden Schritte aus:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
          PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- Stellen Sie den Zugriff auf die Warteschlange SYSTEM.PROTECTION.ERROR.QUEUE ein. Führen Sie dazu die folgenden Schritte aus:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

IBM i Zertifikate und die Keystore-Konfigurationsdatei für AMS unter IBM i einrichten

Ihre erste Aufgabe beim Einrichten des Advanced Message Security-Schutzes ist es, ein Zertifikat zu erstellen und es Ihrer Umgebung zuzuordnen. Die Zuordnung wird über eine Datei konfiguriert, die im Integrated File System (IFS) gehalten wird.

Vorgehensweise

1. Wenn Sie ein selbst signiertes Zertifikat mit dem mit IBM i gelieferten OpenSSL-Tool erstellen möchten, geben Sie den folgenden Befehl aus QShell aus:

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
   $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
   ```

   Der Befehl fordert Sie zur Eingabe verschiedener definierter Namensattribute für ein neues selbst signiertes Zertifikat auf. Hierzu gehören:

   - Allgemeiner Name (CN =)
   - Organisation (O =)
   - Land (C =)

   Dadurch wird ein unverschlüsselter privater Schlüssel und ein übereinstimmender Zertifikat erstellt, und zwar sowohl im PEM-Format (Privacy Enhanced Mail).

   Geben Sie der Einfachheit einfach Werte für den allgemeinen Namen, die Organisation und das Land ein. Diese Attribute und Werte sind wichtig, wenn Sie eine Richtlinie erstellen.

   Zusätzliche Eingabeaufforderungen und Attribute können angepasst werden, indem Sie in der Befehlszeile eine benutzerdefinierte OpenSSL-Konfigurationsdatei mit dem Parameter **-config** angeben. Weitere Informationen zur Syntax der Konfigurationsdatei finden Sie in der OpenSSL-Dokumentation.

Mit dem folgenden Befehl werden beispielsweise zusätzliche Zertifikatserweiterungen für X.509 v3 hinzugefügt:

```
/QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048
-keyout $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365 -config myconfig.cnf
```

Dabei ist 'myconfig.cnf' eine ASCII-Datenstromdatei mit folgenden Inhalten:

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = myextensions

[req_distinguished_name]
countryName = Country Name (2 letter code)
countryName_default = GB
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Hants
localityName = Locality Name (eg, city)
localityName_default = Hursley
organizationName = Organization Name (eg, company)
organizationName_default = IBM United Kingdom
organizationalUnitName = Organizational Unit Name (eg, department)
organizationalUnitName_default = IBM MQ Development
commonName = Common Name (eg, Your Name)

[myextensions]
keyUsage = digitalSignature,nonRepudiation,dataEncipherment,keyEncipherment
extendedKeyUsage = emailProtection
```

2. Für AMS müssen das Zertifikat und der private Schlüssel in der gleichen Datei gespeichert sein. Setzen Sie den folgenden Befehl ab, um dies zu erreichen:

```
cat $HOME/mycert.pem >> $HOME/private.pem
```

Die Datei `private.pem` in $HOME enthält jetzt einen übereinstimmenden privaten Schlüssel und ein Zertifikat, während die Datei `mycert.pem` alle öffentlichen Zertifikate enthält, für die Sie Nachrichten verschlüsseln und Signaturen überprüfen können.

Die beiden Dateien müssen Ihrer Umgebung zugeordnet werden, indem Sie eine Schlüsselspeicherkonfigurationsdatei (`keystore.conf`) in der Standardposition erstellen.

Standardmäßig sucht AMS in einem `.mqs`-Unterverzeichnis Ihres Ausgangsverzeichnisses nach der Schlüsselspeicherkonfiguration.

3. Erstellen Sie in QShell die Datei `keystore.conf` :

```
mkdir -p $HOME/.mqs
echo "pem.private = $HOME/private.pem" > $HOME/.mqs/keystore.conf
echo "pem.public = $HOME/mycert.pem" >> $HOME/.mqs/keystore.conf
echo "pem.password = unused" >> $HOME/.mqs/keystore.conf
```

IBM i *Richtlinie für AMS unter IBM i erstellen*

Bevor Sie eine Richtlinie erstellen, müssen Sie eine Warteschlange erstellen, in der geschützte Nachrichten enthalten sind.

Vorgehensweise

1. Geben Sie an der Eingabeaufforderung Folgendes ein:

```
CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
```

Dabei steht `mqmname` für den Namen Ihres Warteschlangenmanagers.

Überprüfen Sie mit dem Befehl DSPMQM, ob er Warteschlangenmanager in der Lage ist, Sicherheitsrichtlinien zu verwenden. Stellen Sie sicher, dass für **Security Policy Capability** *YES*angezeigt wird.

Die einfachste Richtlinie, die Sie definieren können, ist eine Integritätsrichtlinie, die durch die Erstellung einer Richtlinie mit einem Algorithmus für digitale Signatur, jedoch ohne Verschlüsselungsalgorithmus, erreicht wird.

Nachrichten werden signiert, aber nicht verschlüsselt. Wenn Nachrichten verschlüsselt werden sollen, müssen Sie einen Verschlüsselungsalgorithmus und einen oder mehrere beabsichtigte Nachrichtenempfänger angeben.

Ein Zertifikat im öffentlichen Schlüsselspeicher für einen beabsichtigten Nachrichtenempfänger wird durch einen definierten Namen identifiziert.

2. Zeigen Sie die definierten Namen der Zertifikate im öffentlichen Schlüsselspeicher `mycert.pem` in $HOME an, indem Sie den folgenden Befehl in QShell verwenden:

```
/QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
```

   Sie müssen den definierten Namen als beabsichtigten Empfänger eingeben, und der Richtlinienname muss mit dem Namen der Warteschlange übereinstimmen, der geschützt werden soll.

3. Geben Sie an einer CL-Eingabeaufforderung Folgendes ein, z. B.:

```
SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
O=.., C=..')
```

   Dabei steht `mqmname` für den Namen Ihres Warteschlangenmanagers.

   Sobald die Richtlinie erstellt ist, unterliegen alle Nachrichten, die über diesen Warteschlangennamen eingereiht, durchsucht oder destruktiv entfernt werden, der AMS-Richtlinie.

Zugehörige Verweise

Nachrichten-WS-Manager anzeigen (DSPMQM)

Definieren Sie die MQM-Sicherheitsrichtlinie (SETMQMSPL).

IBM i *Richtlinie für AMS unter IBM i testen*

Verwenden Sie die mit dem Produkt bereitgestellten Musteranwendungen, um Ihre Sicherheitsrichtlinien zu testen.

Informationen zu diesem Vorgang

Mit den mit IBM MQ bereitgestellten Beispielanwendungen wie AMQSPUT4, AMQSGET4 und AMQSGBR4 sowie mit Tools wie WRKMQMMSG können Sie Nachrichten mithilfe des Warteschlangennamens PROTECTED einreihen, durchsuchen und abrufen.

Wenn alles korrekt konfiguriert wurde, sollte es für diesen Benutzer keinen Unterschied im Anwendungsverhalten zu dem Verhalten einer ungeschützte Warteschlange geben.

Ein Benutzer, der nicht für Advanced Message Security konfiguriert ist, oder ein Benutzer, der nicht über den erforderlichen privaten Schlüssel für die Entschlüsselung der Nachricht verfügt, kann die Nachricht jedoch nicht anzeigen. Der Benutzer empfängt einen Beendigungscode von RCFAIL, der äquivalent zu MQCC_FAILED (2) und Ursachencode RC2063 (MQRC_SECURITY_ERROR) ist.

Um zu sehen, dass der AMS-Schutz wirksam ist, stellen Sie einige Testnachrichten in die Warteschlange PROTECTED, z. B. mit AMQSPUT0. Anschließend können Sie eine Aliaswarteschlange erstellen, um die unformatierte geschützte Daten während der Pause zu durchsuchen.

Vorgehensweise

Führen Sie die folgenden Schritte aus, um einem Benutzer die erforderlichen Berechtigungen zu erteilen:

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

Wenn Sie die Verwendung des Namens der ALIAS-Warteschlange verwenden, z. B. mit AMQSBCG4 oder WRKMQMMSG, sollten größere `scrambled` -Nachrichten angezeigt werden, wenn in der Warteschlange PROTECTED Klartextnachrichten angezeigt werden.

Die verwürfelten Nachrichten sind sichtbar, aber der ursprüngliche Klartext kann nicht mit der ALIAS-Warteschlange entschlüsselbar gemacht werden, da es keine Richtlinie für AMS gibt, um die Übereinstimmung mit diesem Namen zu erzwingen. Daher werden die unformatierte geschützte Daten zurückgegeben.

Zugehörige Verweise

Definieren Sie die MQM-Sicherheitsrichtlinie (SETMQMSPL).

Mit MQ-Nachrichten arbeiten (WRKMQMMSG)

Befehls- und Konfigurationereignisse für AMS

Mit Advanced Message Security können Sie Befehle und Konfigurationsereignisnachrichten generieren, die protokolliert werden können und als Datensatz von Richtlinienänderungen für die Prüfung dienen.

Bei den von IBM MQ generierten Befehls und Konfigurationsereignissen handelt es sich um Nachrichten im PCF-Format, die an dedizierte Warteschlangen in dem Warteschlangenmanager gesendet wurden, in dem das Ereignis auftritt.

Die Nachrichten der Konfigurationsereignisse werden an die Warteschlange SYSTEM.ADMIN.CONFIG.EVENT gesendet.

Befehlsereigungsnachrichten werden an die Warteschlange SYSTEM.ADMIN.COMMAND.EVENT gesendet.

Ereignisse werden unabhängig von den Tools generiert, die Sie zum Verwalten der Advanced Message Security-Sicherheitsrichtlinien verwenden.

In Advanced Message Security gibt es vier Ereignistypen, die von verschiedenen Aktionen in Sicherheitsrichtlinien generiert werden:

- „Sicherheitsrichtlinien in AMS erstellen” auf Seite 717, wobei zwei IBM MQ-Ereignisnachrichten generiert werden:
  - Ein Konfigurationsereignis
  - Ein Befehlsereignis
- „Sicherheitsrichtlinien in AMS ändern” auf Seite 718, wobei drei IBM MQ-Ereignisnachrichten generiert werden:
  - Ein Konfigurationsereignis, das alte Sicherheitsrichtlinienwerte enthält
  - Ein Konfigurationsereignis, das neue Sicherheitsrichtlinienwerte enthält.
  - Ein Befehlsereignis
- „Sicherheitsrichtlinien in AMS anzeigen und löschen” auf Seite 719, wobei eine IBM MQ-Ereignisnachricht generiert wird:
  - Ein Befehlsereignis
- „Sicherheitsrichtlinien in AMS entfernen” auf Seite 721, wobei zwei IBM MQ-Ereignisnachrichten generiert werden:
  - Ein Konfigurationsereignis
  - Ein Befehlsereignis

Ereignisprotokollierung für AMS aktivieren und deaktivieren

Sie steuern Befehls-und Konfigurationsereignisse mit den WS-Managerattributen **CONFIGEV** und **CMDEV**. Um diese Ereignisse zu aktivieren, setzen Sie das entsprechende WS-Manager-Attribut auf ENABLED .

Wenn Sie diese Ereignisse inaktivieren möchten, setzen Sie das entsprechende Attribut des Warteschlangenmanagers auf DISABLED .

Vorgehensweise

Konfigurationsereignisse

Wenn Sie Konfigurationsereignisse aktivieren möchten, setzen Sie **CONFIGEV** auf ENABLED . Wenn Sie die Konfigurationsereignisse inaktivieren möchten, setzen Sie **CONFIGEV** auf DISABLED . Sie können beispielsweise Konfigurationsereignisse mit dem folgenden MQSC-Befehl aktivieren:

```
ALTER QMGR CONFIGEV (ENABLED)
```

Befehlsereignisse

Wenn Sie Befehlsereignisse aktivieren möchten, setzen Sie **CMDEV** auf ENABLED . Um Befehlsereignisse für Befehle mit Ausnahme von **DISPLAY MQSC** -Befehlen und Inquire PCF-Befehlen zu aktivieren, setzen Sie **CMDEV** auf NODISPLAY. Um Befehlsereignisse zu inaktivieren, setzen Sie **CMDEV** auf DISABLED. Sie können z. B. Befehlsereignisse mit dem folgenden MQSC-Befehl aktivieren:

```
ALTER QMGR CMDEV (ENABLED)
```

Zugehörige Tasks

Konfigurations-, Befehls- und Protokollierungsereignisse in IBM MQ steuern

Format von Befehlsereignisnachrichten für AMS

Die Befehlsereignisnachricht setzt sich aus den folgenden MQCFH-Struktur-und PCF-Parametern zusammen.

Hier sind die folgenden MQCFH-Werte ausgewählt:

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

Anmerkung: Der Wert für ParameterCount ist zwei, da es immer zwei Parameter des Typs MQCFGR (Gruppe) gibt. Jede Gruppe besteht aus geeigneten Parametern. Die Ereignisdaten bestehen aus zwei Gruppen, CommandContext und CommandData.

CommandContext enthält:

Ereignisbenutzer-ID

Beschreibung:	Die Benutzer-ID, die den Befehl oder Aufruf ausgegeben hat, von dem das Ereignis generiert wurde. (Dies ist die gleiche Benutzer-ID, mit der die Berechtigung zum Absetzen des Befehls oder Aufrufs überprüft wird. Für Befehle, die von einer Warteschlange empfangen werden, ist dies auch die Benutzer-ID (UserIdentifier) aus dem MD der Befehlsnachricht.)
ID:	MQCACF_EVENT_USER_ID.
Datentyp:	MQCFST.
Maximale Länge:	MQ_USER_ID_LENGTH.
Zurückgegeben:	Immer.

EventOrigin

Beschreibung:	Der Ursprung der Aktion, die das Ereignis ausgelöst hat.

ID: MQIACF_EVENT_ORIGIN.

Datentyp: MQCFIN.

Werte:

MQEVO_CONSOLE
Konsolbefehl-Befehlszeile.

MQEVO_MSG
Befehlsnachrichten vom IBM MQ Explorer-Plug-in

Zurückgegeben: Immer.

EventQMgr

Beschreibung: Der Warteschlangenmanager, in den der Befehl oder Aufruf eingegeben wurde. (Der Warteschlangenmanager, in dem der Befehl ausgeführt wird und das das Ereignis generiert, befindet sich im MD der Ereignisnachricht.)

ID: MQCACF_EVENT_Q_MGR.

Datentyp: MQCFST.

Maximale Länge: MQ_Q_MGR_NAME_LENGTH.

Zurückgegeben: Immer.

Ereignisabrechnungstoken

Beschreibung: Für Befehle, die als Nachricht (MQEVO_MSG) empfangen werden, das Abrechnungstoken (AccountingToken) von der MD-Nachricht der Befehlsnachricht.

ID: MQBACF_EVENT_ACCOUNTING_TOKEN.

Datentyp: MQCFBS.

Maximale Länge: MQ_ACCOUNTING_TOKEN_LENGTH.

Zurückgegeben: Nur wenn EventOrigin MQEVO_MSG ist.

Ereignisidentitätsdaten

Beschreibung: Für Befehle, die als Nachricht (MQEVO_MSG) empfangen wurden, Anwendungsidentitätsdaten (ApplIdentityData) aus dem MD der Befehlsnachricht.

ID: MQCACF_EVENT_APPL_IDENTITY.

Datentyp: MQCFST.

Maximale Länge: MQ_APPL_IDENTITY_DATA_LENGTH.

Zurückgegeben: Nur wenn EventOrigin MQEVO_MSG ist.

EventApplType

Beschreibung: Für Befehle, die als Nachricht (MQEVO_MSG) empfangen wurden, der Typ der Anwendung (PutApplType) aus dem MD der Befehlsnachricht.

ID: MQIACF_EVENT_APPL_TYPE.

Datentyp: MQCFIN.

Zurückgegeben: Nur wenn EventOrigin MQEVO_MSG ist.

Ereignisanwendungsname

Beschreibung: Für Befehle, die als Nachricht (MQEVO_MSG) empfangen wurden, der Name der Anwendung (PutApplName) aus dem MD der Befehlsnachricht.

ID: MQCACF_EVENT_APPL_NAME.

Datentyp: MQCFST.

Maximale Länge: MQ_APPL_NAME_LENGTH.

Zurückgegeben: Nur wenn EventOrigin MQEVO_MSG ist.

EventApplOrigin

Beschreibung: Für Befehle, die als Nachricht (MQEVO_MSG) empfangen werden, die Anwendungsursprungsdaten (ApplOriginData) aus dem MD der Befehlsnachricht.

ID: MQCACF_EVENT_APPL_ORIGIN.

Datentyp: MQCFST.

Maximale Länge: MQ_APPL_ORIGIN_DATA_LENGTH.

Zurückgegeben: Nur wenn EventOrigin MQEVO_MSG ist.

Befehl

Beschreibung: Der Befehlscode.

ID: MQIACF_COMMAND.

Datentyp: MQCFIN.

Werte: **MQCMD_INQUIRE_PROT_POLICY, numerischer Wert 205**
MQCMD_CREATE_PROT_POLICY, numerischer Wert 206
MQCMD_DELETE_PROT_POLICY, numerischer Wert 207
MQCMD_CHANGE_PROT_POLICY, numerischer Wert 208
Diese sind in IBM MQ 8.0 `cmqcfc.h` definiert.

Zurückgegeben: Immer.

CommandData enthält PCF-Elemente, die den PCF-Befehl enthalten.

Format von Konfigurationsereignisnachrichten für AMS

Konfigurationsereignisse sind PCF-Nachrichten im Advanced Message Security-Standardformat.

Mögliche Werte für den MQMD-Nachrichtendeskriptor finden Sie in Ereignisnachricht MQMD (Nachrichtendeskriptor) .

Die folgenden MQMD-Werte sind ausgewählt:

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR           //for both CLI and command server
```

Der Nachrichtenpuffer setzt sich aus der MQCFH-Struktur und der darauf folgenden Parameterstruktur zusammen. Mögliche MQCFH-Werte finden Sie in Ereignisnachricht MQCFH (PCF-Header) .

Hier sind die folgenden MQCFH-Werte ausgewählt:

```
Type = MQCFT_EVENT
Command = MQCMD_CONFIG_EVENT
MsgSeqNumber = 1 or 2          // 2 will be in case of Change Object event
Control = MQCFC_LAST or MQCFC_NOT_LAST     //MQCFC_NOT_LAST will be in case of 1 Change Object
event
ParameterCount = reflects number of PCF parameters following MQCFH
CompCode = MQCC_WARNING
Reason = one of {MQRC_CONFIG_CREATE_OBJECT, MQRC_CONFIG_CHANGE_OBJECT, MQRC_CONFIG_DELETE_OB
JECT}
```

Folgende Parameter werden nach MQCFH verwendet:

EventUserID

Beschreibung: Die Benutzer-ID, die den Befehl oder Aufruf ausgegeben hat, von dem das Ereignis generiert wurde. (Dies ist die gleiche Benutzer-ID, mit der die Berechtigung zum Absetzen des Befehls oder Aufrufs überprüft wird. Für Befehle, die von einer Warteschlange empfangen werden, ist dies auch die Benutzer-ID (UserIdentifier) aus dem MD der Befehlsnachricht.)

ID: **MQCACF_EVENT_USER_ID**

Datentyp: MQCFST.

Maximale Länge: MQ_USER_ID_LENGTH.

Zurückgegeben: Immer.

SecurityId

Beschreibung: Wert von MQMD.AccountingToken im Fall einer Befehlsservernachricht oder Windows-SID für lokalen Befehl.

ID: **MQBACF_EVENT_SECURITY_ID**

Datentyp: MQCBS.

Maximale Länge: MQ_SECURITY_ID_LENGTH.

Zurückgegeben: Immer.

EventOrigin

Beschreibung: Der Ursprung der Aktion, die das Ereignis ausgelöst hat.

ID: **MQIACF_EVENT_ORIGIN**

Datentyp: MQCFIN.

Werte:

MQEVO_CONSOLE
Konsolbefehl-Befehlszeile.

MQEVO_MSG
Befehlsnachricht aus dem IBM MQ Explorer-Plug-in.

Zurückgegeben: Immer.

EventQMgr

Beschreibung: Der Warteschlangenmanager, in den der Befehl oder Aufruf eingegeben wurde. (Der Warteschlangenmanager, in dem der Befehl ausgeführt wird und das das Ereignis generiert, befindet sich im MD der Ereignisnachricht.)

ID: **MQCACF_EVENT_Q_MGR**

Datentyp: MQCFST

Maximale Länge: MQ_Q_MGR_NAME_LENGTH

Zurückgegeben: Immer.

ObjectType

Beschreibung: Objekttyp.

ID: **MQIACF_OBJECT_TYPE**

Datentyp: MQCFIN

Wert: **MQOT_PROT_POLICY**
Advanced Message Security-Schutzrichtlinie. **1019** - numerischer Wert, der in IBM MQ 8.0 oder in der Datei cmqc.h definiert ist.

Zurückgegeben: Immer.

PolicyName

Beschreibung: Der Name der Advanced Message Security-Richtlinie.

ID: **MQCA_POLICY_NAME** .

Datentyp: MQCFST.

Wert: **2112** - ein numerischer Wert, der in IBM MQ 8.0 oder in der cmqc.h-Datei definiert ist.

Maximale Länge: MQ_OBJECT_NAME_LENGTH.

Zurückgegeben: Immer.

PolicyVersion

Beschreibung: Die Version der Advanced Message Security-Richtlinie.

ID: **MQIA_POLICY_VERSION**

Datentyp: MQCFIN

Wert **238** - ein numerischer Wert, der in IBM MQ 8.0 oder in der cmqc.h-Datei definiert ist.

Zurückgegeben: Immer

TolerateFlag

Beschreibung: Flag für die Toleranz der Advanced Message Security-Richtlinie

ID: **MQIA_TOLERATE_UNPROTECTED**

Datentyp: MQCFIN

Wert **235** - ein numerischer Wert, der in IBM MQ 8.0 oder in der cmqc.h-Datei definiert ist.

Zurückgegeben: Immer.

SignatureAlgorithm

Beschreibung: Der Algorithmus der Advanced Message SecurityRichtliniensignatur.

ID: **MQIA_SIGNATURE_ALGORITHM**

Datentyp: MQCFIN

Wert: **236** - ein numerischer Wert, der in IBM MQ 8.0 oder in der cmqc.h-Datei definiert ist.

Zurückgegeben: Sobald in der Advanced Message Security-Richtlinie ein Signaturalgorithmus definiert ist

EncryptionAlgorithm

Beschreibung: Der Verschlüsselungsalgorithmus der Advanced Message Security-Richtlinie.

ID: **MQIA_ENCRYPTION_ALGORITHM**

Datentyp: MQCFIN

Wert: **237** - ein numerischer Wert, der in IBM MQ 8.0 oder in der `cmqc.h`-Datei definiert ist.

Zurückgegeben: Sobald in der IBM MQ-Richtlinie ein Verschlüsselungsalgorithmus definiert ist

SignerDNs

Beschreibung: Subjekt DistinguishedName der zulässigen Unterzeichner.

ID: **MQCA_SIGNER_DN**

Datentyp: MQCFSL

Wert: **2113** - ein numerischer Wert, der in IBM MQ 8.0 oder in der `cmqc.h`-Datei definiert ist.

Maximale Länge: Längster Unterzeichner-DN in der Richtlinie, aber nicht mehr als MQ_DISTINGUISHED_NAME_LENGTH

Zurückgegeben: Bei jeder Definition in der IBM MQ-Richtlinie.

RecipientDNs

Beschreibung: Subjekt DistinguishedName der zulässigen Unterzeichner.

ID: **MQCA_RECIPIENT_DN**

Datentyp: MQCFSL

Wert: **2114** - ein numerischer Wert, der in IBM MQ 8.0 oder in der `cmqc.h`-Datei definiert ist.

Maximale Länge: Längster Empfänger-DN in der Richtlinie, aber nicht mehr als MQ_DISTINGUISHED_NAME_LENGTH.

Zurückgegeben: Bei jeder Definition in der IBM MQ-Richtlinie.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Dokumentation ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe
IBM Europe, Middle East and Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Défense
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Lizenzierung von geistigem Eigentum

IBM Japan, Ltd.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in dieser Veröffentlichung werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East and Africa
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Die in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Um diese so realistisch wie möglich zu gestalten, enthalten sie auch Namen von Personen, Firmen, Marken und Produkten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musterprogramme, die in Quellensprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos (d. h. ohne Zahlung an IBM) kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

Informationen zu Programmierschnittstellen

Die bereitgestellten Informationen zur Programmierschnittstelle sollen Sie bei der Erstellung von Anwendungssoftware für dieses Programm unterstützen.

Dieses Handbuch enthält Informationen über vorgesehene Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zu schreiben, um die Services von WebSphere MQzu erhalten.

Diese Informationen können jedoch auch Angaben über Diagnose, Bearbeitung und Optimierung enthalten. Die Informationen zu Diagnose, Bearbeitung und Optimierung sollten Ihnen bei der Fehlerbehebung für die Anwendungssoftware helfen.

Wichtig: Verwenden Sie diese Diagnose-, Änderungs-und Optimierungsinformationen nicht als Programmierschnittstelle, da sie Änderungen unterliegen.

Marken

IBM, das IBM Logo, ibm.com, sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein.

Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (https://www.eclipse.org/) entwickelt wurde.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Teilenummer:

(1P) P/N: