7.5

IBM WebSphere MQ schützen

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen” auf Seite 343 gelesen werden.

Diese Ausgabe bezieht sich auf Version 7 Release 5 von IBM® WebSphere MQ und auf 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

Sicherheit

Sicherheit ist ein wichtiger Aspekt sowohl für Entwickler von IBM WebSphere MQ-Anwendungen als auch für Systemadministratoren, die IBM WebSphere MQ-Berechtigungen konfigurieren.

Sicherheit - Übersicht

In dieser Themensammlung werden die IBM WebSphere 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 WebSphere MQ implementiert sind.

Sicherheitskonzepte und -mechanismen

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

Die allgemein akzeptierten Sicherheitsaspekte sind wie folgt:

- „Identifikation und Authentifizierung” auf Seite 5
- „Berechtigung” auf Seite 6
- „Prüfprotokollierungs” auf Seite 6
- „Vertraulichkeit” auf Seite 7
- „Datenintegrität” auf Seite 7

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 7
- „Nachrichtendigests und digitale Signaturen” auf Seite 9
- „Digitale Zertifikate” auf Seite 10
- „Public Key Infrastructure (PKI)” auf Seite 14

Wenn Sie eine IBM WebSphere MQ -Implementierung planen, überlegen Sie, welche Sicherheitsmechanismen erforderlich sind, um die für Sie wichtigen Sicherheitsaspekte zu implementieren. Informationen dazu, was Sie nach dem Lesen dieser Themen beachten sollten, finden Sie unter „Sicherheitsanforderungen planen” auf Seite 49.

Zugehörige Konzepte

„Zwei Warteschlangenmanager über SSL oder TLS verbinden” auf Seite 218
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.

„Mit SSL oder TLS arbeiten” auf Seite 117
In diesen Abschnitten finden Sie Anweisungen für die Ausführung von einzelnen Tasks im Zusammenhang mit der Verwendung von SSL oder TLS mit IBM WebSphere MQ.

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 WebSphere MQ-Umgebung um einen relevanten Sicherheitsservice, da IBM WebSphere 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 WebSphere MQ mit IBM WebSphere MQ Advanced Message Security stellt keinen Nichtrepudiationsservice als Teil seiner Basisfunktion bereit. Diese Produktdokumentation enthält jedoch Vorschläge, wie Sie Ihren eigenen fälschungssicheren Herkunftsnachweis in einer WebSphere MQ -Umgebung bereitstellen können, indem Sie eigene Exitprogramme schreiben.

Zugehörige Konzepte

„Identifikation und Authentifizierung in IBM WebSphere MQ” auf Seite 22
In IBM WebSphere 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 WebSphere MQ” auf Seite 22
Sie können die Berechtigung verwenden, um zu begrenzen, was bestimmte Einzelpersonen oder Anwendungen in Ihrer IBM WebSphere MQ -Umgebung tun können.

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 52 und den zugehörigen Unterabschnitten.

Zugehörige Konzepte

„Prüfung in IBM WebSphere MQ” auf Seite 23
IBM WebSphere 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 für WebSphere MQbeschrieben.

Der Begriff *Entität* wird verwendet, um auf einen Warteschlangenmanager, einen WebSphere MQ MQI-Client, einen einzelnen Benutzer oder ein anderes System zu verweisen, das Nachrichten austauschen kann.

Zugehörige Konzepte

„Verschlüsselung in IBM WebSphere MQ” auf Seite 24
In IBM WebSphere MQ erfolgt die Verschlüsselung mit Hilfe der Protokolle Secure Sockets Layer (SSL) und Transport Security Layer (TLS).

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* bezeichnet (manchmal *Verschlüsselung*).
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* bezeichnet (manchmal *Entschlüsselung*).

Eine Definition der Verschlüsselung finden Sie im Glossar .

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 und TLS" auf Seite 20.

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 gemeinsamen Schlüssel verwenden, werden als *symmetrische* Algorithmen bezeichnet.Abbildung 1 auf Seite 8 zeigt die Verschlüsselung symmetrischer Schlüssel.
- 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 *asymmetrische* Algorithmen bezeichnet.Abbildung 2 auf Seite 8 veranschaulicht die asymmetrische Verschlüsselung, die auch als *Verschlüsselung mit öffentlichem Schlüssel*bezeichnet wird.

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

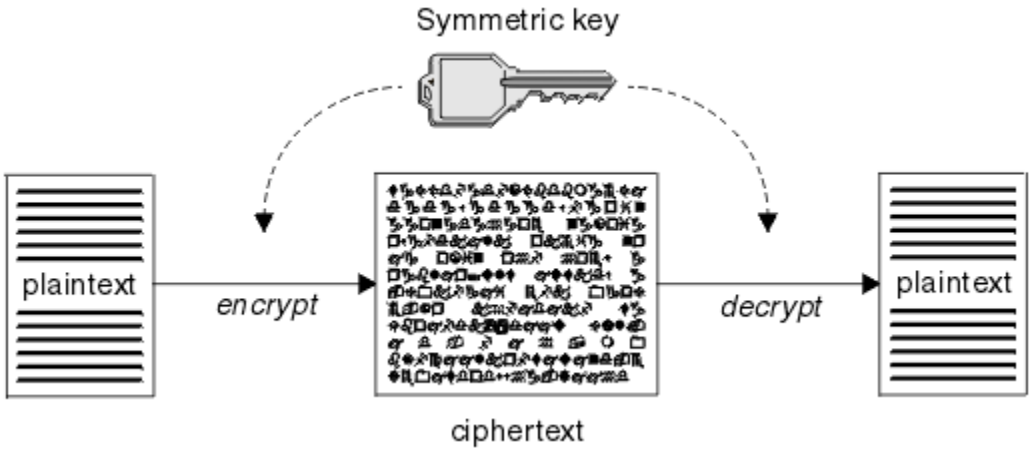


Abbildung 1. Symmetrische Schlüsselkryptografie

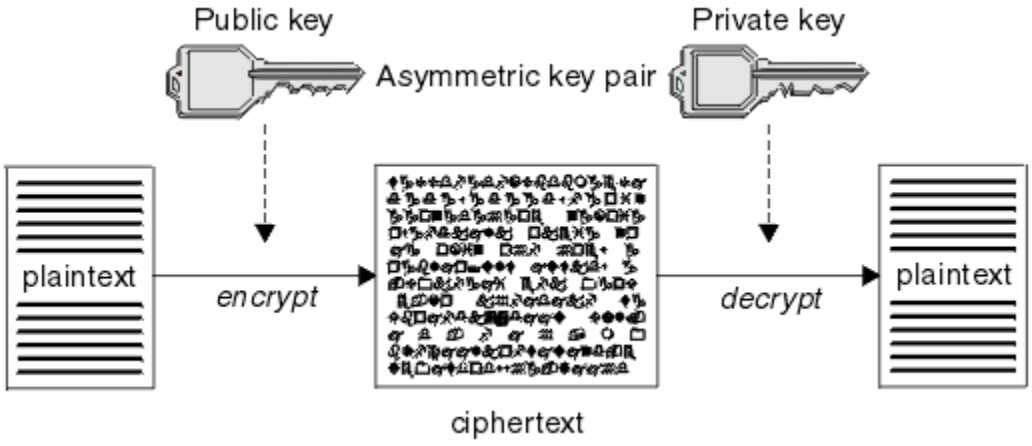


Abbildung 2. Asymmetrische Schlüsselkryptografie

Abbildung 2 auf Seite 8 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 fester Größe des Inhalts einer Nachricht, die durch eine Hashfunktion berechnet wird. Ein Message-Digest kann verschlüsselt werden und eine digitale Signatur bilden.

Die Größe von Nachrichten ist von Natur aus variabel. Ein Nachrichten-Digest ist eine numerische Darstellung des Inhalts einer Nachricht mit fester Größe. Ein Nachrichten-Digest wird durch eine Hashfunktion berechnet, bei der es sich um eine Transformation handelt, die zwei Kriterien erfüllt:

- Die Hashfunktion muss unidirektional 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 und TLS” auf Seite 20

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 11 beschrieben.

Was ist in einem digitalen Zertifikat?

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

Digitale Zertifikate, die von WebSphere MQ verwendet werden, entsprechen dem Standard X.509 , der die erforderlichen Informationen und das Format für das Senden 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. Eine nicht kritische Erweiterung kann hingegen vom System ignoriert werden, wenn es die Erweiterung 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
WebSphere MQ unterstützt digitale Zertifikate, die dem Standard X.509 entsprechen. Sie erfordert die Clientauthentifizierungsoption.

Da es sich bei IBM WebSphere MQ um ein Peer-to-Peer-System handelt, wird es im Sinne der SSL-Terminologie als Clientauthentifizierung angesehen. Daher muss jedes für die SSL-Authentifizierung verwendete persönliche Zertifikat eine Schlüsselnutzung 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 in IBM WebSphere MQ finden Sie in „Zertifikatsprüfrichtlinien in IBM WebSphere MQ” auf Seite 35.

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 159.
- 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.

Die folgenden Attributtypen werden häufig im DN gefunden:

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

Zugehörige Konzepte

„Was ist in einem digitalen Zertifikat?" auf Seite 10
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 normalerweise von dem Zertifikatsmanagementtool generiert, das Ihr System verwendet, z. B. vom Tool iKeyman auf UNIX-, Linux®-und Windows -Systemen und von 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 13 veranschaulicht den Prozess, mit dem ein digitales Zertifikat von einer Zertifizierungsstelle abgerufen wird.

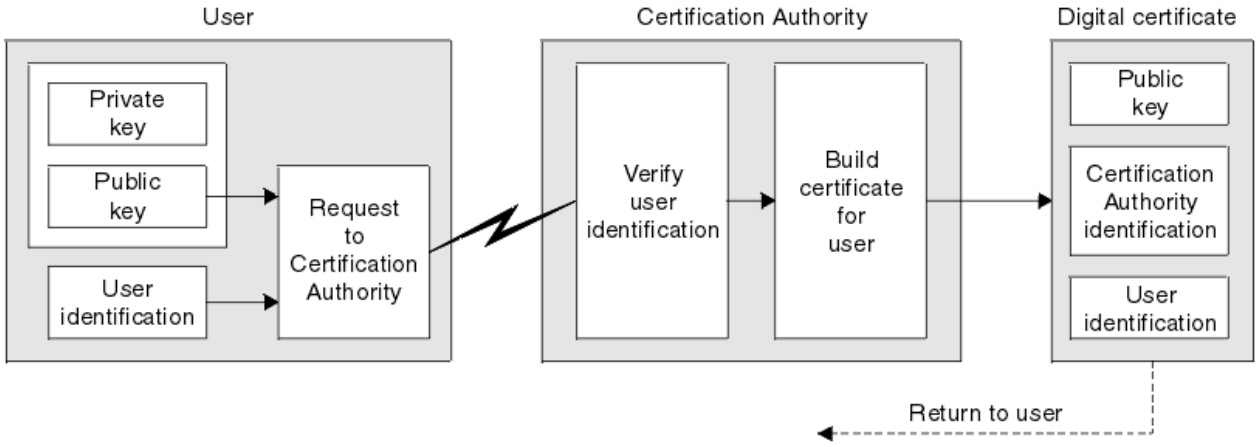


Abbildung 3. Abrufen eines digitalen Zertifikats

Im Diagramm:

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

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 10.

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, auch *Zertifizierungspfad* genannt, ist eine Liste von Zertifikaten, die zur 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 14 zeigt einen Zertifizierungspfad vom Zertifikateigner bis zur Stamm-CA, wo die Vertrauenskette beginnt.

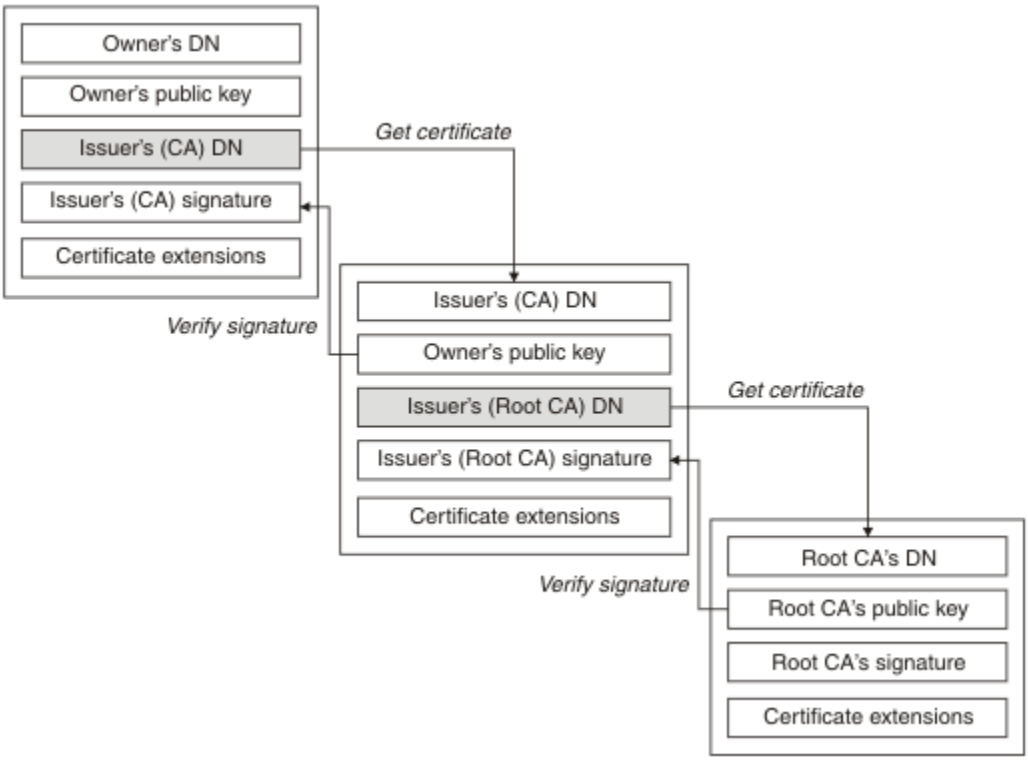


Abbildung 4. Kette der Vertrauenskette

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.

Eine Definition des Zertifikatverfalls finden Sie im Glossar .

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

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

WebSphere MQ kann prüfen, ob ein Zertifikat widerrufen wurde, indem eine Anforderung an einen OCSP-Responder (Online Certificate Status Protocol) gesendet wird (nur auf UNIX-, Linux -und Windows -Systemen). Alternativ können sie auf eine CRL auf einem LDAP-Server zugreifen. Die OCSP-Widerrufs-und CRL-Informationen werden von einer Zertifizierungsstelle veröffentlicht. Weitere Informationen finden Sie in „Mit widerrufenden Zertifikaten arbeiten” auf Seite 159.

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 10. RAs prüfen, ob 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: SSL und 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 WebSphere MQ unterstützt sowohl SSL als auch TLS.

Die primären Ziele beider Protokolle sind die Gewährleistung von Vertraulichkeit (manchmal auch als *Datenschutz*bezeichnet), Datenintegrität, Identifikation und Authentifizierung mithilfe 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

„Sicherheitsprotokolle in IBM WebSphere MQ" auf Seite 24
IBM WebSphere MQ unterstützt sowohl das TLS-Protokoll (Transport Layer Security) als auch das SSL-Protokoll (Secure Sockets Layer), um die Sicherheit auf Verbindungsebene für Nachrichtenkanäle und MQI-Kanäle bereitzustellen.

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) concepts

Die Protokolle SSL und TLS ermöglichen es zwei Parteien, sich untereinander zu identifizieren und zu authentifizieren und unter Gewährleistung von Vertraulichkeit und Datenintegrität miteinander zu kommunizieren. Das TLS-Protokoll wurde vom Netscape SSL 3.0-Protokoll entwickelt, aber TLS und SSL sind nicht interaktiv.

Die SSL- und TLS-Protokolle ermöglichen 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 SSL- oder TLS-Verbindung wird von einer Anwendung aufgebaut, die dabei zum SSL- oder TLS-Client wird. Die Anwendung, die Empfänger der Verbindung ist, wird zum SSL- oder TLS-Server. Jede neue Sitzung beginnt mit einem Handshake, der durch das Protokoll SSL oder TLS definiert ist.

Eine vollständige Liste der von IBM WebSphere MQ unterstützten CipherSpecs finden Sie unter „CipherSpecs angeben" auf Seite 230.

Weitere Informationen zum SSL-Protokoll finden Sie auf der Webseite https://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt. 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- bzw. TLS-Handshake ermöglicht dem SSL/TLS-Client und dem SSL/TLS-Server die Erstellung der geheimen Schlüssel, die sie für die Kommunikation verwenden.

Dieser Abschnitt enthält eine Zusammenfassung der Schritte, mit denen der SSL-oder TLS-Client und der 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. SSL oder 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.

Hier eine Übersicht der Schritte in Zusammenhang mit dem SSL-Handshake:

1. Der SSL- oder TLS-Client sendet eine Nachricht vom Typ "client hello", in der kryptographische Informationen aufgeführt sind, wie z. B. die SSL- bzw. TLS-Version und die vom Client unterstützten CipherSuites (in der vom Client vorgegebenen Reihenfolge). 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 SSL/TLS-Server antwortet mit einer Nachricht vom Typ "server hello", welche die vom Server aus der vom Client bereitgestellten Liste ausgewählte CipherSuite enthält sowie die Sitzungs-ID und eine weitere wahlfreie Bytefolge. 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 SSL- oder TLS-Client überprüft das digitale Zertifikat des Servers. Weitere Informationen finden Sie im Abschnitt „Wie SSL und TLS Identifikation, Authentifizierung, Vertraulichkeit und Integrität bereitstellen” auf Seite 17.
4. Der SSL/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 SSL- oder 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 fehlendes digitales Zertifikat". Dieser Alert ist nur eine Warnung, aber bei einigen Implementierungen schlägt der Handshake fehl, wenn die Clientauthentifizierung obligatorisch ist.
6. Der SSL- oder TLS-Server überprüft das Clientzertifikat. Weitere Informationen finden Sie im Abschnitt „Wie SSL und TLS Identifikation, Authentifizierung, Vertraulichkeit und Integrität bereitstellen” auf Seite 17.
7. Der SSL- oder TLS-Client sendet dem Server eine "Fertigstellungs"-Nachricht, die mit dem geheimen Schlüssel verschlüsselt ist und anzeigt, dass der clientseitige Teil des Handshake abgeschlossen ist.
8. Der SSL- oder TLS-Server sendet dem Client eine "Fertigstellungs"-Nachricht, die mit dem geheimen Schlüssel verschlüsselt ist und anzeigt, dass der serverseitige Teil des Handshake abgeschlossen ist.
9. Für die Dauer der SSL- bzw. TLS-Sitzung können Server und Client nun Nachrichten austauschen, die symmetrisch mit dem gemeinsam genutzten geheimen Schlüssel verschlüsselt sind.

Abbildung 5 auf Seite 17 veranschaulicht den SSL- oder TLS-Handshake.

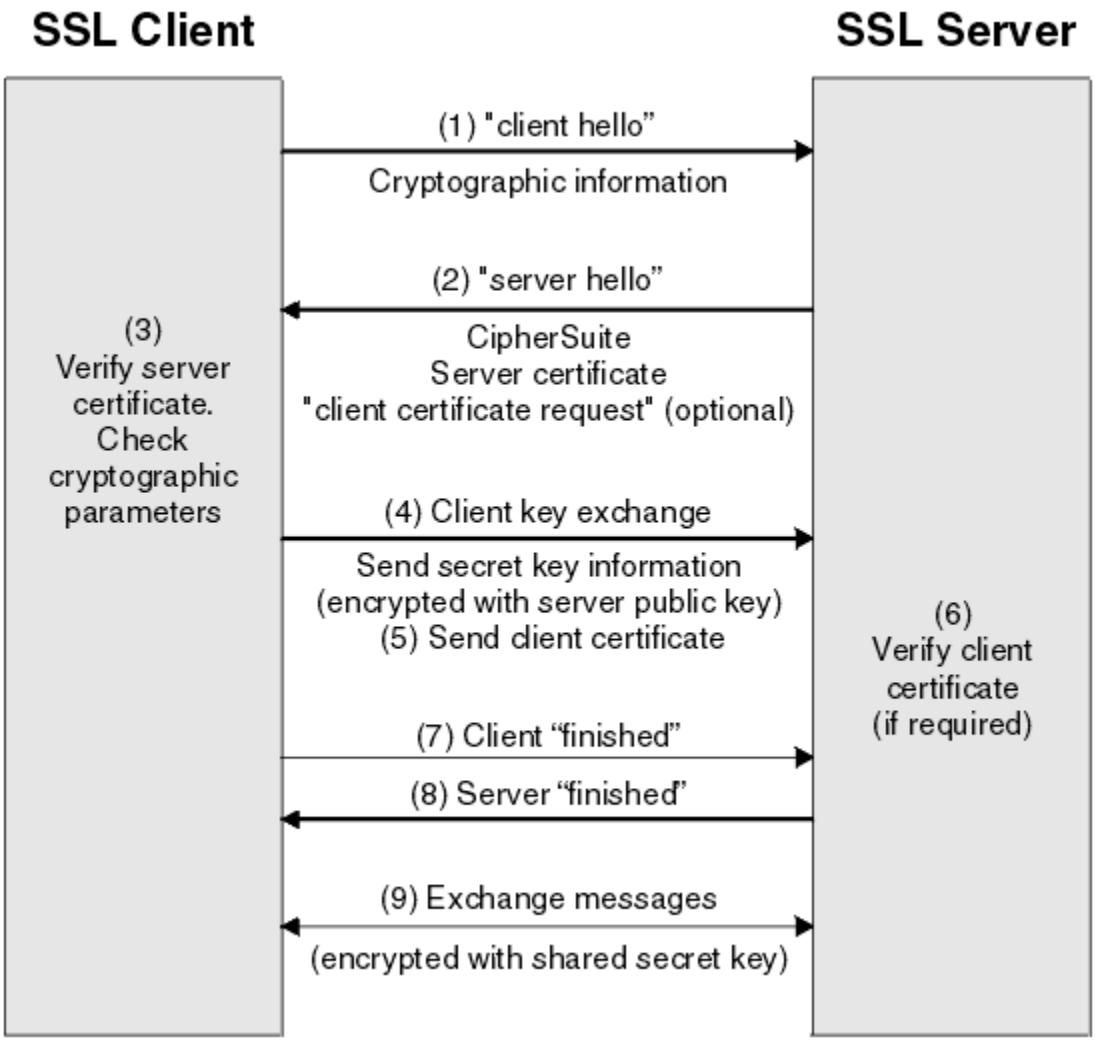


Abbildung 5. Übersicht über den SSL- oder TLS-Handshake

Wie SSL und TLS Identifikation, Authentifizierung, Vertraulichkeit und Integrität bereitstellen

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 16.

Wie SSL und TLS Authentifizierung bereitstellen

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.

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 16 des Handshake sendet. Der Austausch abgeschlossener Nachrichten, die mit dem geheimen Schlüssel verschlüsselt wurden (Schritte „7" auf Seite 16 und „8" auf Seite 16 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 von digitalen Zertifikaten während des SSL- oder TLS-Handshakes ist Teil des Authentifizierungsprozesses. Weitere Informationen darüber, wie Zertifikate Schutz vor der Ipersonation bieten, finden Sie in den zugehörigen Informationen. Folgende Zertifikate sind erforderlich, wobei Zertifizierungsstelle X das Zertifikat an den SSL- oder TLS-Client und Zertifizierungsstelle Y das Zertifikat an den SSL- oder TLS-Server ausstellt:

Soll nur eine Serverauthentifizierung durchgeführt werden, benötigt der SSL- oder TLS-Server:

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

und der SSL- oder TLS-Client benötigt:

- Das CA-Zertifikat für Zertifizierungsstelle Y

Wenn für den SSL-oder TLS-Server eine Clientauthentifizierung erforderlich ist, 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 SSL- oder TLS-Server als auch der Client benötigen unter Umständen weitere Zertifikate von Zertifizierungsstellen, um eine Zertifikatskette zum Zertifikat der Rootzertifizierungsstelle zu bilden. Weitere Informationen zu Zertifikatsketten finden Sie in den zugehörigen Informationen.

Was während der Zertifikatsprüfung passiert

Wie bereits in den Schritten „3” auf Seite 16 und „6” auf Seite 16 der Übersicht erwähnt, prüft der SSL- oder TLS-Client das Zertifikat des Servers und der SSL- oder TLS-Server prüft das Zertifikat des Clients. Für diese Überprüfung gibt es vier Aspekte:

1. Die digitale Signatur wird geprüft (siehe „Digitale Signaturen in SSL und TLS” auf Seite 20).
2. Die Zertifikatskette wird geprüft. Sie sollten über temporäre CA-Zertifikate verfügen (siehe „Funktionsweise der Zertifikatsketten” auf Seite 13).
3. Das Verfallsdatum und die Gültigkeitsdauer werden überprüft.
4. Der Widerrufsstatus des Zertifikats wird überprüft (siehe „Mit widerrufenden Zertifikaten arbeiten” auf Seite 159).

Geheimer Schlüssel zurückgesetzt

Während eines SSL- oder TLS-Handshakes wird ein *geheimer Schlüssel* generiert, um Daten zwischen dem SSL- oder TLS-Client und -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 im Abschnitt „Nachrichtendigests und digitale Signaturen” auf Seite 9.

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 SSL und TLS Vertraulichkeit bereitstellen

SSL und TLS verwenden eine Kombination aus symmetrischer und asymmetrischer Verschlüsselung, um Nachrichtenvertraulichkeit zu gewährleisten. Während des SSL- oder TLS-Handshakes vereinbaren der SSL- oder TLS-Client und -Server einen Verschlüsselungsalgorithmus und einen gemeinsam genutzten geheimen Schlüssel, der nur für eine Sitzung verwendet wird. Alle zwischen dem SSL- oder TLS-Client und -Server übertragenen Nachrichten werden mit dem vereinbarten Algorithmus und Schlüssel verschlüsselt. So wird sichergestellt, dass die Nachricht auch dann vertraulich bleibt, wenn sie abgefangen wird. SSL unterstützt eine Vielzahl von Verschlüsselungsalgorithmen. Da SSL und TLS beim Transport des gemeinsam genutzten geheimen Schlüssels eine asymmetrische Verschlüsselung verwenden, gibt es kein Schlüsselverteilungsproblem. Weitere Informationen zu Verschlüsselungsverfahren finden Sie in „Kryptografie” auf Seite 7.

Wie SSL und TLS Integrität bereitstellen

SSL und TLS stellen Integrität durch Berechnung eines Message-Digest bereit. Weitere Informationen finden Sie in „Datenintegrität von Nachrichten” auf Seite 240.

Sowohl SSL als auch TLS gewährleisten die Datenintegrität, sofern die CipherSpec der Kanaldefinition, wie in der Tabelle im Abschnitt „CipherSpecs angeben” auf Seite 230 beschrieben, einen Hashalgorithmus verwendet.

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- oder SSL-Verbindung müssen sich auf dieselbe CipherSpec einigen, damit eine Kommunikation möglich ist.

Wichtig: Bei der Bearbeitung von IBM WebSphere 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 angeben” auf Seite 230.

Bei einer CipherSuite handelt es sich um eine Reihe von Verschlüsselungsalgorithmen, die für eine SSL- oder 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.

Für jede Komponente der Suite stehen mehrere Optionen zur Auswahl; für eine TLS- oder SSL-Verbindung sind allerdings nur bestimmte Kombinationen möglich. Der Name einer gültigen CipherSuite definiert die Kombination der verwendeten Algorithmen. So gibt die CipherSuite SSL_RSA_WITH_RC4_128_MD5 Folgendes an:

- Der Algorithmus für RSA-Schlüsselaustausch und -Authentifizierung
- Den RC4-Verschlüsselungsalgorithmus mit einem 128-Bit-Schlüssel
- Den MAC-Algorithmus MD5

Für den Schlüsselaustausch und die Authentifizierung stehen zwar mehrere Algorithmen zur Verfügung, der RSA-Algorithmus ist jedoch der derzeit gängigste. Außerdem gibt es eine Vielzahl von Algorithmen für die Datenverschlüsselung und MAC.

Digitale Signaturen in SSL und 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 20 veranschaulicht diesen Prozess.

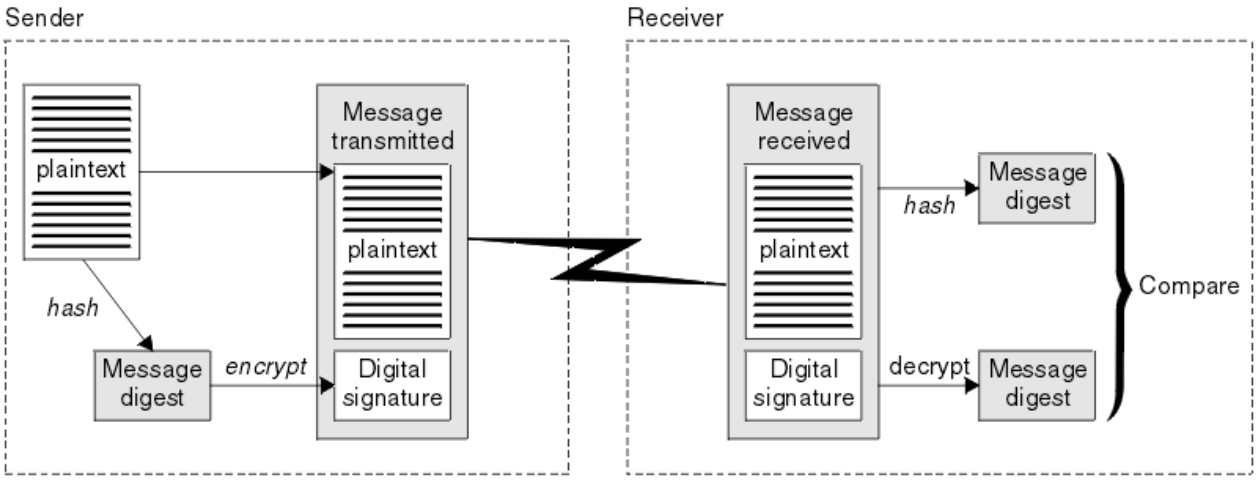


Abbildung 6. Der Prozess der digitalen Signatur

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.

IBM WebSphere 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.

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 Version 1.2. Bei der Konfiguration für Suite B-konforme Operationen wird nur die oben aufgeführte eingeschränkte Gruppe von Verschlüsselungsalgorithmen verwendet.
2. Ein Übergangsprofil für die Verwendung mit TLS Version 1.0 oder TLS Version 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.

Unter Windowskönnen UNIX -und Linux -Systeme, WebSphere MQ, so konfiguriert werden, dass sie dem Suite B-konformen TLS 1.2 -Profil entsprechen. Das Suite B-Übergangsprofil wird jedoch nicht unterstützt. Weitere Informationen finden Sie in „NSA Suite B-Verschlüsselung in IBM WebSphere MQ" auf Seite 32.

Zugehörige Informationen

„Federal Information Processing Standards" auf Seite 21
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 WebSphere MQ Sicherheitsmechanismen

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

IBM WebSphere MQ stellt Mechanismen zur Implementierung aller Sicherheitskonzepte bereit, die in „Sicherheitskonzepte und -mechanismen” auf Seite 5eingeführt wurden. Diese werden in den folgenden Abschnitten ausführlicher behandelt.

Identifikation und Authentifizierung in IBM WebSphere MQ

In IBM WebSphere 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 WebSphere 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 5
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 WebSphere MQ

Sie können die Berechtigung verwenden, um zu begrenzen, was bestimmte Einzelpersonen oder Anwendungen in Ihrer IBM WebSphere MQ -Umgebung tun können.

Es folgen einige Beispiele für die Berechtigung in einer IBM WebSphere MQ -Umgebung:

- Nur einem berechtigten Administrator das Absetzen von Befehlen zur Verwaltung von IBM WebSphere MQ -Ressourcen erlauben
- 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 52 und den zugehörigen Unterabschnitten.

Zugehörige Konzepte

„Berechtigung” auf Seite 6

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 WebSphere MQ

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

Nachfolgend sind einige Beispiele für die Prüfung in einer IBM WebSphere MQ -Umgebung zu finden:

- 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 6
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 WebSphere MQ

Sie können die Vertraulichkeit in IBM WebSphere MQ durch Verschlüsseln von Nachrichten implementieren.

Die folgenden Beispiele zeigen, wie die Vertraulichkeit in einer IBM WebSphere MQ -Umgebung gewährleistet werden kann:

- Nachdem ein sendender Nachrichtenkanalagent (MCA) eine Nachricht aus einer Übertragungswarteschlange abgerufen hat, verschlüsselt IBM WebSphere MQ die Nachricht mit SSL oder TLS, bevor sie über das Netz an den empfangenden MCA 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 WebSphere MQ bereitgestellte Verfahren zur Zugriffssteuerung aus, um die Inhalte vor nicht autorisierter Offenlegung zu schützen. Für ein höheres Maß an Sicherheit können Sie jedoch IBM WebSphere MQ Advanced Message Security verwenden, um die in den Warteschlangen gespeicherten Nachrichten zu verschlüsseln.

Zugehörige Konzepte

„Vertraulichkeit" auf Seite 7
Der Service *Vertraulichkeit* schützt sensible Informationen vor unbefugter Offenlegung.

Datenintegrität in IBM WebSphere MQ

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

Die folgenden Beispiele zeigen, wie die Datenintegrität in einer IBM WebSphere MQ -Umgebung sichergestellt werden kann:

- Sie können SSL oder TLS verwenden, um festzustellen, ob der Inhalt einer Nachricht absichtlich geändert wurde, während er über ein Netz übertragen wurde. In SSL und TLS werden durchlaufende geänderte Nachrichten mit Hilfe des Nachrichtenauszugsalgorithmus erkannt. Alle IBM WebSphere MQ CipherSpecs stellen einen Message-Digest-Algorithmus bereit, mit Ausnahme von TLS_RSA_WITH_NULL_NULL, der keine Nachrichtendatenintegrität bereitstellt.

- Während Nachrichten in einer lokalen Warteschlange gespeichert werden, können die von IBM WebSphere MQ bereitgestellten Zugriffssteuerungsmechanismen als ausreichend betrachtet werden, um eine absichtliche Änderung der Nachrichteninhalte zu verhindern. Für ein höheres Maß an Sicherheit können Sie jedoch IBM WebSphere MQ 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.

Zugehörige Konzepte

„Datenintegrität" auf Seite 7
Der *Datenintegritätsdienst* stellt fest, ob unbefugte Änderungen an Daten vorgenommen wurden.

Verschlüsselung in IBM WebSphere MQ

In IBM WebSphere MQ erfolgt die Verschlüsselung mit Hilfe der Protokolle Secure Sockets Layer (SSL) und Transport Security Layer (TLS).

Weitere Informationen finden Sie in „Sicherheitsprotokolle in IBM WebSphere MQ" auf Seite 24.

Zugehörige Konzepte

„Verschlüsselungskonzepte" auf Seite 7
In dieser Themensammlung werden die Konzepte der Verschlüsselung für WebSphere MQbeschrieben.

Sicherheitsprotokolle in IBM WebSphere MQ

IBM WebSphere MQ unterstützt sowohl das TLS-Protokoll (Transport Layer Security) als auch das SSL-Protokoll (Secure Sockets Layer), um die Sicherheit auf Verbindungsebene für Nachrichtenkanäle und MQI-Kanäle bereitzustellen.

Nachrichtenkanäle und MQI-Kanäle können das SSL- oder TLS-Protokoll verwenden, um die Sicherheit auf Verbindungsebene bereitzustellen.Ein aufrufender Nachrichtenkanalagent ist in diesem Fall der SSL- oder TLS-Client, und der Nachrichtenkanalagent, der auf den Aufruf reagiert, ist der SSL- oder TLS-Server. WebSphere MQ unterstützt Version 3.0 des SSL-Protokolls und Version 1.0 und Version 1.2 des Protokolls Transport Layer Security (TLS). Sie geben die Verschlüsselungsalgorithmen an, die von SSL oder Protokoll verwendet werden, indem Sie eine CipherSpec als Teil der Kanaldefinition angeben.

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 SSL- oder TLS-Handshake sendet der MCA das digitale Zertifikat des Warteschlangenmanagers an seinen Partner-MCA am anderen Ende des Kanals. Der WebSphere MQ -Code am Clientende eines MQI-Kanals agiert für den Benutzer der WebSphere MQ -Clientanwendung. Während des SSL-oder TLS-Handshakes sendet der WebSphere MQ -Code das digitale Zertifikat des Benutzers an den MCA am Serverende des MQI-Kanals.

Warteschlangenmanager und WebSphere MQ -Clientbenutzer müssen keine persönlichen digitalen Zertifikate haben, wenn sie als SSL-oder TLS-Clients agieren, es sei denn, auf der Serverseite des Kanals ist SSLCAUTH (REQUIRED) angegeben.

Digitale Zertifikate werden in einem *Schlüsselrepository* gespeichert. Das Warteschlangenmanagerattribut *SSLKeyRepository* gibt die Position des Schlüsselrepositorys an, das das digitale Zertifikat des Warteschlangenmanagers manager enthält. Auf einem WebSphere MQ -Clientsystem gibt die Umgebungsvariable MQSSLKEYR die Position des Schlüsselrepositorys an, das das digitale Zertifikat des Benutzers enthält. Alternativ kann eine WebSphere MQ -Clientanwendung ihre Position im Feld *KeyRepository* der Struktur der SSL-und TLS-Konfigurationsoptionen (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.

Zugehörige Konzepte

„Verschlüsselte Sicherheitsprotokolle: SSL und TLS" auf Seite 15
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 WebSphere MQ unterstützt sowohl SSL als auch TLS.

IBM WebSphere MQ Unterstützung für SSL und TLS

IBM WebSphere MQ unterstützt sowohl das SSL-Protokoll (Secure Sockets Layer) als auch das TLS-Protokoll (Transport Layer Security).

Weitere Informationen zu den SSL-und TLS-Protokollen finden Sie in den zugehörigen Informationen.

IBM WebSphere MQ stellt die folgende Unterstützung für SSL Version 3.0 und TLS 1.0 und TLS 1.2:

Java-und JMS-Clients
Diese Clients stellen SSL- und TLS-Unterstützung über JVM zur Verfügung.

UNIX, Linux, and Windows-und HP Integrity NonStop Server -Systeme
Für UNIX, Linux, and Windows-und HP Integrity NonStop Server -Systeme wird die SSL-und TLS-Unterstützung mit IBM WebSphere MQinstalliert.

Informationen zu Voraussetzungen für die SSL-und TLS-Unterstützung von IBM WebSphere MQ finden Sie in Systemvoraussetzungen für IBM WebSphere MQ.

Das SSL- oder TLS-Schlüsselrepository
Eine gegenseitig authentifizierte SSL-oder TLS-Verbindung erfordert ein Schlüsselrepository (das auf verschiedenen Plattformen unter verschiedenen Namen bekannt sein kann) an jedem Ende der Verbindung. 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. Die folgenden Geschäftsnamen werden auf den Plattformen und Umgebungen verwendet, die SSL und TLS unterstützen:

Java und JMS	Keystore und Truststore
Windows UNIX Linux Windows UNIX Linux Windows , UNIX and Linux -Systeme	Schlüsseldatenbankdatei

Weitere Informationen finden Sie unter „Digitale Zertifikate” auf Seite 10 und „Secure Sockets Layer (SSL) and Transport Layer Security (TLS) concepts” auf Seite 15.

Für eine gegenseitig authentifizierte SSL- oder TLS-Verbindung ist an jedem Ende der Verbindung ein Schlüsselrepository erforderlich. Das Schlüsselrepository kann Folgendes 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. Jedem Warteschlangenmanager oder WebSphere MQ MQI-Client wird ein separates persönliches Zertifikat zugeordnet. Persönliche Zertifikate sind für einen SSL- oder 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 WebSphere MQ-Schlüsselrepositorys schützen” auf Seite 26.

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

Windows UNIX Linux **Windows, UNIX and Linux -Systeme**
Auf Windows-Systemen ist UNIX and Linux das Schlüsselrepository eine Schlüsseldatenbankdatei. Der Name der Schlüsseldatenbankdatei muss eine Dateierweiterung von `.kdb` haben. Unter UNIX and Linuxlautet die Standardschlüsseldatenbankdatei für Warteschlangenmanager `QM1` beispielsweise `/var/mqm/qmgrs/QM1/ssl/key.kdb` . Wenn IBM WebSphere MQ im Standardverzeichnis in-

stalliert ist, lautet der entsprechende Pfad unter Windows `C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\key.kdb`.

Unter Windows , UNIX and Linux -Systemen verfügt jede Schlüsseldatenbankdatei über eine zugehörige Kennwortstashdatei. 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: Auf Windows- UNIX and Linux -Systemen können PKCS #11 -Verschlüsselungshardwarekarten die Zertifikate und Schlüssel enthalten, die andernfalls in einer Schlüsseldatenbankdatei enthalten sind. Wenn Zertifikate und Schlüssel auf PKCS #11 -Karten gespeichert werden, benötigt WebSphere MQ weiterhin Zugriff auf eine Schlüsseldatenbankdatei und eine Kennwortstashdatei.

Auf Windows -und UNIX -Systemen enthält die Schlüsseldatenbank auch den privaten Schlüssel für das persönliche Zertifikat, das dem Warteschlangenmanager oder dem WebSphere MQ MQI-Client zugeordnet ist.

IBM WebSphere MQ-Schlüsselrepositorys schützen
Das Schlüsselrepository für IBM WebSphere MQ ist 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 WebSphere 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.

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 WebSphere MQ Explorer aktualisieren. Weitere Informationen finden Sie unter Befehl MQCMD_REFRESH_SECURITY und im Abschnitt *SSL- oder TLS-Sicherheit aktualisieren* im Abschnitt WebSphere MQ Explorer dieser Produktdokumentation.

Zugehörige Konzepte
„Clientansicht des SSL-Schlüsselrepositoryinhalts und der SSL-Einstellungen neu anzeigen” auf Seite 26
Wenn Sie die Clientanwendung mit dem aktualisierten Inhalt des Schlüsselrepositorys aktualisieren möchten, müssen Sie die Clientanwendung stoppen und erneut starten.

Clientansicht des SSL-Schlüsselrepositoryinhalts und der SSL-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 WebSphere MQ -Client nicht aktualisieren. Es gibt kein Äquivalent des Befehls REFRESH SECURITY TYPE (SSL) für Clients (siehe 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 26
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.

Federal Information Processing Standards (FIPS)

Dieser Abschnitt enthält eine Einführung in das FIPS-Verschlüsselungsprogramm (Federal Information Processing Standards) Cryptomodule Validation Program des US National Institute of Standards and Technology und die Verschlüsselungsfunktionen, die in SSL-oder TLS-Kanälen für Windows-, UNIX and Linux-und z/OS -Systeme verwendet werden können.

Die FIPS 140-2-Konformität einer IBM WebSphere MQ SSL-oder TLS-Verbindung auf UNIX-, Linux-und Windows -Systemen finden Sie hier „Federal Information Processing Standards (FIPS) für UNIX, Linuxund Windows" auf Seite 27.

Wenn Verschlüsselungshardware vorhanden ist, werden von IBM WebSphere 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 WebSphere 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 115
Erstellen Sie Ihre Schlüsselrepositorys mit FIPS-konformer Software und geben Sie dann an, dass der Kanal FIPS-zertifizierte CipherSpecs verwenden muss.

„iKeyman, iKeycmd, runmqakm und runmqckm verwenden" auf Seite 120
Auf UNIX-, Linux -und Windows -Systemen können Sie Schlüssel und digitale Zertifikate über die iKeyman -GUI oder über die Befehlszeile mit iKeycmd oder runmqakm verwalten.

Zugehörige Tasks

SSL in WebSphere MQ-Klassen für Java aktivieren

Secure Sockets Layer (SSL) mit WebSphere MQ-Klassen für JMS verwenden

Zugehörige Verweise

SSL-Eigenschaften von JMS-Objekten

Zugehörige Informationen

„Federal Information Processing Standards" auf Seite 21
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).

Federal Information Processing Standards (FIPS) für UNIX, Linuxund Windows

Wenn die Verschlüsselung auf einem SSL-oder TLS-Kanal auf Windows-oder UNIX and Linux -Systemen erforderlich ist, verwendet WebSphere MQ ein Verschlüsselungspaket namens IBM Crypto for C (ICC). Auf

den Plattformen Windows, UNIX and Linux , hat die ICC-Software das FIPS-Verschlüsselungsprogramm (Federal Information Processing Standards) Cryptomodule Validation Program des US National Institute of Standards and Technology auf Stufe 140-2 bestanden.

Für die Konformität gemäß FIPS 140-2 einer mit WebSphere MQ hergestellten SSL- oder TLS-Verbindung auf Windows- und UNIX and Linux-Systemen gilt Folgendes:

- Für alle IBM WebSphere 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 WebSphere MQ MQI-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 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 WebSphere MQ -Klassen für Java-Anwendungen, die den Clientmodus verwenden, verwendet die Verbindung die SSL-und TLS-Implementierungen der JRE und ist FIPS-konform, wenn die folgenden Bedingungen erfüllt sind:
  - Die zur Ausführung der Anwendung verwendete Java Runtime Environment ist mit der installierten Betriebssystemversion und Hardwarearchitektur FIPS-konform.
  - Sie haben angegeben, dass nur FIPS-zertifizierte Verschlüsselung verwendet werden soll, wie im zugehörigen Abschnitt für den Java-Client beschrieben.
  - Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .
- Für IBM WebSphere MQ -Klassen für JMS-Anwendungen, die den Clientmodus verwenden, verwendet die Verbindung die SSL-und TLS-Implementierungen der JRE und ist FIPS-konform, wenn die folgenden Bedingungen erfüllt sind:
  - Die zur Ausführung der Anwendung verwendete Java Runtime Environment ist mit der installierten Betriebssystemversion und Hardwarearchitektur FIPS-konform.
  - Sie haben angegeben, dass nur FIPS-zertifizierte Verschlüsselung verwendet werden soll, wie im entsprechenden Thema für den JMS-Client beschrieben.
  - Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .
- Für nicht verwaltete .NET-Clientanwendungen verwendet die Verbindung Global Security Kit (GSKit). Sie ist mit FIPS konform, wenn folgende 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 im entsprechenden Thema für den .NET-Client beschrieben.
  - Alle Schlüsselrepositorys wurden ausschließlich mit FIPS-konformer Software erstellt und bearbeitet, z. B. **runmqakm** mit der Option `-fips` .
- Für nicht verwaltete XMS .NET-Clientanwendungen verwendet die Verbindung Global Security Kit (GSKit). Sie ist mit FIPS konform, wenn folgende 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 AIX-, Linux-, HP-UX-, Solaris-, Windows-und z/OS -Plattformen sind FIPS 140-2-zertifiziert, außer wie in der Readme-Datei, die in jedem Fixpack oder Refresh-Pack enthalten ist, angegeben.

Bei SSL- und TLS-Verbindungen, die GSKit verwenden, hat die FIPS 140-2-zertifizierte Komponente die Bezeichnung *ICC*. 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 WebSphere MQ die vom Hardwarehersteller bereitgestellten Verschlüsselungsmodul verwendet. Ist dies der Fall, ist die Konfiguration nur FIPS-konform, wenn die Verschlüsselungsmodule FIPS-zertifiziert sind.

Anmerkung: Die für mit FIPS 140-2-konformen Betrieb konfigurierten 32 Bit Solaris x86-SSL- und TLS-Clients schlagen fehl, wenn sie auf Intel-Systemen ausgeführt werden. Dieser Fehler tritt auf, weil die FIPS 140-2-konforme GSKit-Crypto Solaris x86 32-Bit-Bibliotheksdatei auf dem Intel-Chipsatz nicht geladen werden kann. Auf betroffenen Systemen wird der Fehler AMQ9655 im Clientfehlerprotokoll aufgeführt. Sie können dieses Problem beheben, indem Sie die FIPS 140-2-Konformität inaktivieren oder die Clientanwendung für 64 Bit kompilieren, da 64-Bit-Code nicht betroffen ist.

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

Wenn WebSphere MQ für den Betrieb in Übereinstimmung mit FIPS 140-2 konfiguriert wird, werden zusätzliche Einschränkungen in Bezug auf Triple DES (3DES) CipherSpecs erzwungen. 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. Da der geheime Sitzungsschlüssel nicht standardmäßig von WebSphere MQ zurückgesetzt wird, muss dieser Vorgang ausdrücklich konfiguriert werden. 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 gehei-

men Schlüsseln finden Sie im Abschnitt „Zurücksetzen von geheimen SSL-und TLS-Schlüsseln” auf Seite 237.

WebSphere 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 115
Erstellen Sie Ihre Schlüsselrepositorys mit FIPS-konformer Software und geben Sie dann an, dass der Kanal FIPS-zertifizierte CipherSpecs verwenden muss.

„iKeyman, iKeycmd, runmqakm und runmqckm verwenden” auf Seite 120
Auf UNIX-, Linux -und Windows -Systemen können Sie Schlüssel und digitale Zertifikate über die iKeyman -GUI oder über die Befehlszeile mit iKeycmd oder runmqakm verwalten.

Zugehörige Tasks

SSL in WebSphere MQ-Klassen für Java aktivieren

Secure Sockets Layer (SSL) mit WebSphere MQ-Klassen für JMS verwenden

Zugehörige Verweise

SSL-Eigenschaften von JMS-Objekten

Zugehörige Informationen

„Federal Information Processing Standards” auf Seite 21
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).

SSL und TLS auf dem IBM WebSphere MQ MQI-Client

IBM WebSphere MQ unterstützt SSL und TLS auf Clients. Die Verwendung von SSL oder TLS kann auf verschiedene Weisen angepasst werden.

IBM WebSphere MQ bietet SSL-und TLS-Unterstützung für IBM WebSphere MQ MQI-Clients auf Systemen mit Windowsund UNIX and Linux . Wenn Sie IBM WebSphere MQ Classes for Java verwenden, lesen Sie den Abschnitt Using WebSphere MQ classes for Java . Wenn Sie IBM WebSphere MQ Classes for JMS verwenden, lesen Sie den Abschnitt Using WebSphere MQ classes for JMS. Der Rest dieses Abschnitts gilt nicht für die Java-oder JMS-Umgebungen.

Sie können das Schlüsselrepository für einen IBM WebSphere MQ MQI-Client entweder mit dem Wert MQSSLKEYR in Ihrer IBM WebSphere MQ -Clientkonfigurationsdatei angeben oder wenn Ihre Anwendung einen MQCONNX-Aufruf ausführt. Es gibt drei Möglichkeiten für die Angabe, dass ein Kanal SSL verwendet:

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

Die Verwendung von SSL in einem Kanal kann nicht über die Umgebungsvariable MQSERVER angegeben werden.

Sie können Ihre vorhandenen IBM WebSphere MQ MQI-Clientanwendungen weiterhin ohne SSL ausführen, solange SSL am anderen Ende des Kanals nicht angegeben ist.

Werden auf einem Clientsystem Änderungen an den Inhalten oder der Adresse des SSL-Schlüsselrepositorys, der Authentifizierungsinformationen oder den Parametern der Verschlüsselungshardware vorgenommen, müssen alle SSL-Verbindungen beendet werden, damit die Änderungen in den Clientverbindungskanälen übernommen werden, die die Anwendung für die Verbindung zum WS-Manager verwendet. Starten Sie die SSL-Kanäle erneut, nachdem alle Verbindungen getrennt wurden. Alle neuen SSL-Einstel-

lungen werden jetzt verwendet. Diese Einstellungen entsprechen den Einstellungen, die mit dem Befehl REFRESH SECURITY TYPE (SSL) auf WS-Managersystemen aktualisiert werden.

Wenn Ihr IBM WebSphere MQ MQI-Client auf einem Windows-, UNIX and Linux -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 geheimer SSL-Schlüssel und FIPS werden auf IBM WebSphere MQ MQI-Clients unterstützt. Weitere Informationen hierzu finden Sie unter „Zurücksetzen von geheimen SSL-und TLS-Schlüsseln” auf Seite 237 und „Federal Information Processing Standards (FIPS) für UNIX, Linuxund Windows” auf Seite 27.

Weitere Informationen zur SSL-Unterstützung für IBM WebSphere MQ MQI-Clients finden Sie im Abschnitt „IBM WebSphere MQ MQI-Clientsicherheit einrichten” auf Seite 114 .

Zugehörige Tasks

Client mit einer Konfigurationsdatei konfigurieren

Für einen MQI-Kanal SSL festlegen

Damit SSL 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 WebSphere 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 WebSphere MQ MQI-Clientanwendung 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 Windowsverwenden.

   Auf Windows -Systemen können Sie den Steuerbefehl **setmqscp** verwenden, um die Clientverbindungskanaldefinitionen in Active Directoryzu veröffentlichen. Eine oder mehrere dieser Definitionen können den Namen einer Verschlüsselungsspezifikation (CipherSpec) angeben.

Wenn eine Clientanwendung beispielsweise eine Clientverbindungskanaldefinition in einer MQCD-Struktur in einem MQCONNX-Aufruf bereitstellt, wird diese Definition vor allen Einträgen in einer Clientkanaldefinitionstabelle verwendet, auf die der WebSphere MQ -Client zugreifen kann.

Mit der Umgebungsvariablen MQSERVER kann auf der Clientseite keine Kanaldefinition für einen MQI-Kanal, der SSL verwendet, bereitgestellt werden.

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 WebSphere MQ MQI-Client angeben” auf Seite 236

Sie haben drei Optionen zur Angabe einer CipherSpec für einen IBM WebSphere MQ MQI-Client.

CipherSpecs und CipherSuites in IBM WebSphere MQ
IBM WebSphere MQ unterstützt SSL-und TLS- CipherSpecssowie RSA-und Diffie-Hellman-Algorithmen.

WebSphere MQ unterstützt SSL V3 und TLS V1.0 und V1.2 CipherSpecs.

WebSphere MQ unterstützt die Schlüsselaustausch-und Authentifizierungsalgorithmen RSA und Diffie-Hellman. Die Größe des Schlüssels, der während des SSL-Handshakes verwendet wird, kann vom verwendeten digitalen Zertifikat abhängen, aber einige CipherSpecs enthalten eine Spezifikation der Größe des Handshakeschlüssels. Größere Handshake-Schlüsselgrößen bieten eine stärkere Authentifizierung. Bei kleineren Schlüsselgrößen ist der Handshake schneller.

Zugehörige Konzepte
„CipherSpecs und CipherSuites" auf Seite 19
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 WebSphere MQ
Dieser Abschnitt enthält Informationen zur Konfiguration von IBM WebSphere MQ auf Windows-, Linux- und UNIX-Systemen für die Konformität mit dem 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 WebSphere 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 von IBM WebSphere MQ Version 7.5 wird die Version von Suite B aufgelistet, die von den einzelnen Produktverwaltungsstufen umgesetzt wird. Wenn Sie IBM WebSphere MQ für die Umsetzung der Suite B-Konformität konfigurieren, ziehen Sie bei der Planung der Anwendung von Wartungspaketen immer die Readme-Datei zu Rate (siehe IBM MQ, WebSphere MQund MQSeries-Produkt-Readme-Dateien).

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

Tabelle 1. 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 WebSphere MQ nicht mit dem Suite B-Standard konform ist, kann IBM WebSphere MQ so konfiguriert werden, dass es einer oder beiden Sicherheitsstufen auf Windows-, UNIX -und Linux -Systemen entspricht. Direkt nach der erfolgreichen Konfiguration von IBM WebSphere 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 WebSphere MQ CipherSpecsfinden Sie unter „CipherSpecs angeben” auf Seite 230 .

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 WebSphere MQ für die Verwendung von Zertifikaten konfiguriert werden, deren digitaler Signaturalgorithmus und öffentlicher Schlüsseltyp gemäß der konfigurierten Suite B-Sicherheitsstufe des fernen Partners zulässig sind. 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 Tools **iKeycmd** und **iKeyman** unterstützen die Erstellung digitaler Zertifikate für den Suite B-konformen Betrieb nicht.

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 auf Systemen mit UNIX, Linux, and Windows erstellen” auf Seite 129.

Informationen zum Anfordern eines von einer Zertifizierungsstelle signierten digitalen Zertifikats für die Produktionsverwendung in Suite B finden Sie in „Persönliches Zertifikat unter UNIX, Linux, and Windows anfordern” auf Seite 131.

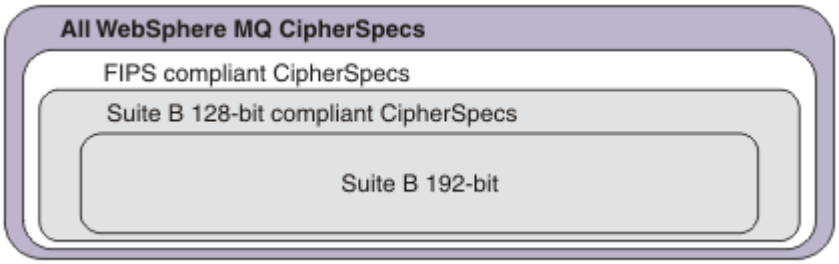
Anmerkung: Die verwendete Zertifizierungsstelle muss digitale Zertifikate generieren, die die in der IETF-RFC 6460 beschriebenen Anforderungen erfüllen.

FIPS 140-2 und Suite B

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 WebSphere MQ für den FIPS

140-2-konformen Betrieb konfiguriert wurde. Es ist daher möglich, WebSphere MQ für die FIPS-und Suite B-Konformität gleichzeitig zu konfigurieren. In diesem Fall gelten beide Gruppen von Einschränkungen.

Das folgende Diagramm zeigt die Beziehung zwischen diesen Untergrup-



pen:

WebSphere MQ für Suite B-konformen Betrieb konfigurieren

Informationen zur Konfiguration von IBM WebSphere MQ unter Windows, UNIX und Linux für Suite B-konforme Operationen finden Sie unter „IBM WebSphere MQ für Suite B konfigurieren” auf Seite 34.

IBM WebSphere MQ unterstützt Suite B-konforme Operationen auf den Plattformen IBM i und z/OS nicht. Die WebSphere MQ -Java-und -JMS-Clients unterstützen auch keine Suite B-konformen Operationen.

Zugehörige Konzepte

„Angeben, dass nur FIPS-zertifizierte CipherSpecs während der Ausführung auf dem MQI-Client verwendet werden” auf Seite 115
Erstellen Sie Ihre Schlüsselrepositorys mit FIPS-konformer Software und geben Sie dann an, dass der Kanal FIPS-zertifizierte CipherSpecs verwenden muss.

IBM WebSphere MQ für Suite B konfigurieren
IBM WebSphere MQ kann für den Betrieb in Übereinstimmung mit dem NSA Suite B-Standard auf UNIX, Linux, and Windows -Systemen konfiguriert werden.

Suite B beschränkt die Gruppe aktivierter Verschlüsselungsalgorithmen, um eine sichere Sicherheitsstufe zu gewährleisten.IBM WebSphere MQ kann für den Betrieb in Übereinstimmung mit Suite B konfiguriert werden, um eine höhere Sicherheitsstufe bereitzustellen. Weitere Informationen zu Suite B finden Sie in „National Security Agency (NSA) Suite B Cryptography” auf Seite 21. Weitere Informationen zur Suite B-Konfiguration und ihre Auswirkungen auf SSL- und TLS-Kanäle finden Sie im Abschnitt „NSA Suite B-Verschlüsselung in IBM WebSphere MQ” auf Seite 32.

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.

MQI-Client

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

1. Durch Festlegen des Feldes **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. Setzen Sie das Attribut **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 unter SSL-Zeilengruppe der Clientkonfigurationsdatei .

Weitere Informationen zur Verwendung der Umgebungsvariablen MQSUITEB finden Sie im Abschnitt 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 WebSphere MQ Classes for .NET finden Sie unter MQEnvironment .NET-Klasse.

Zertifikatsprüfrichtlinien in IBM WebSphere 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 JSSE-Komponente der Java Runtime Environment ab. Weitere Informationen zur Validierungsrichtlinie für Zertifikate finden Sie in der Dokumentation zu Ihrer JRE.
- Für Systeme mit UNIX, 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.

Informationen zur Konfiguration der Richtlinie für die Zertifikatsprüfung auf UNIX, Linux, and Windows-Systemen finden Sie im Abschnitt „Zertifikatsprüfrichtlinien in IBM WebSphere MQ konfigurieren” auf Seite 36. Weitere Informationen zu den Unterschieden zwischen den Basis-und Standardzertifikatsprüf-

richtlinien finden Sie unter Certificate validation and trust policy design on UNIX, Linux and Windows systems .

Zertifikatsprüfrichtlinien in IBM WebSphere MQ konfigurieren
Sie können auf vier Arten angeben, welche SSL/TLS-Zertifikatsprüfrichtlinie verwendet wird, um digitale Zertifikate, die von fernen Partnersystemen empfangen werden, auf Gültigkeit zu prüfen.

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 unter 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 im Abschnitt MQCERTVPOL .
3. Verwenden Sie die Einstellung des Parameters für die Optimierung von Client-SSL-Einstellungen, *CertificateValPolicy* Weitere Informationen zur Verwendung dieser Einstellung enthält die SSL-Zeilengruppe der Clientkonfigurationsdatei .

Weitere Informationen zu Zertifikaten für Zertifikatvalidierungen finden Sie in „Zertifikatsprüfrichtlinien in IBM WebSphere MQ” auf Seite 35.

Digitale Zertifikate und CipherSpec -Kompatibilität in IBM WebSphere 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 WebSphere MQ erläutert.

In früheren Releases von IBM WebSphere MQverwendeten alle unterstützten SSL-und TLS- CipherSpecs den RSA-Algorithmus für digitale Signaturen und Schlüsselvereinbarung. Alle unterstützten Typen von digitalen Zertifikaten waren mit allen unterstützten CipherSpecs kompatibel, so dass es möglich war, die CipherSpec für einen beliebigen Kanal zu ändern, ohne digitale Zertifikate ändern zu müssen.

In IBM WebSphere MQ v7.5 kann nur eine Untergruppe der unterstützten CipherSpecs mit allen unterstützten Typen digitaler Zertifikate 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, auf dem TLS 1.2-Protokoll basierende CipherSpecs verwenden möchten, stellen Sie sicher, dass die digitalen Zertifikate nicht den MD5-Algorithmus in ihren digitalen Signaturen verwenden. Diese Einschränkung gilt nicht für ältere CipherSpecs, die die Protokolle SSL 3.0 und TLS 1.0 verwenden. Sie können weiterhin Zertifikate mit digitalen Signaturen des Typs MD5 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
```

Dabei steht `cert_label` für die Zertifikatsbezeichnung des Algorithmus für digitale Signatur, den Sie anzeigen müssen.

Anmerkung: Obwohl das Tool **iKeycmd (runmqckm)** und die grafische Benutzerschnittstelle von **iKeyman (strmqikm)** verwendet werden können, um eine Auswahl von Algorithmen für digitale Signaturen anzuzeigen, stellt das Tool **runmqakm** einen größeren Bereich bereit.

Die Ausführung des Befehls **runmqakm** erzeugt eine Ausgabe, die die Verwendung des angegebenen Signaturalgorithmus anzeigt:

```
Label : ibmwebspheremqexample
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

Es können nicht alle CipherSpecs mit allen digitalen Zertifikaten verwendet werden. Es gibt drei Typen von CipherSpec, die mit dem Namenspräfix "CipherSpec" gekennzeichnet sind. Jeder Typ von CipherSpec beschränkt sich auf die Art des digitalen Zertifikats, das verwendet werden kann. Diese Einschränkungen gelten für alle SSL-und TLS-Verbindungen von WebSphere MQ , sind jedoch besonders für Benutzer der Elliptic Curve-Verschlüsselung relevant.

Die Beziehungen zwischen CipherSpecs und digitalen Zertifikaten werden in der folgenden Tabelle zusammengefasst:

Tabelle 2. 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 Signaturalgorithmen für die Elliptic Curve verwenden.	RSA	RSA	ECDHE
3	(Alle anderen)	CipherSpecs, die öffentliche RSA-Schlüssel und digitale RSA-Signaturalgorithmen verwenden.	RSA	RSA	RSA

Anmerkung: Typ 1 und 2 CipherSpecs werden nur von WebSphere MQ -Warteschlangenmanagern und MQI-Clients auf den UNIX, Linux, and Windows -Plattformen 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.

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 können verschiedene digitale Signaturalgorithmen verwendet werden, u. a. "RSA mit MD5" oder "ECDSA mit SHA-256". 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.

Geben Sie 3 CipherSpecs ein, dass das persönliche Zertifikat einen öffentlichen RSA-Schlüssel haben 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 WebSphere MQ beispielsweise so konfiguriert ist, dass es den Standards FIPS 140-2 oder NSA Suite B entspricht, wird auch der Bereich zulässiger Konfigurationen begrenzt. Weitere Informationen finden Sie im folgenden Abschnitt.

Ein WebSphere MQ WS-Manager kann nur ein einzelnes persönliches Zertifikat verwenden, um sich selbst zu identifizieren. Dies bedeutet, dass alle Kanäle auf dem Warteschlangenmanager dasselbe digitale Zertifikat verwenden. Daher kann jeder Warteschlangenmanager jeweils nur einen Typ von CipherSpec verwenden. Ebenso kann eine WebSphere MQ -Clientanwendung nur ein einzelnes persönliches Zertifikat verwenden, um sich selbst zu identifizieren. Dies bedeutet, dass alle SSL-und TLS-Verbindungen innerhalb eines einzelnen Anwendungsprozesses dasselbe digitale Zertifikat verwenden. Daher kann jeder Clientanwendungsprozess jeweils nur einen Typ von CipherSpec verwenden.

Die drei CipherSpec-Typen arbeiten nicht direkt zusammen: dies ist eine Einschränkung der aktuellen SSL- und TLS-Standards. Angenommen, Sie haben die ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec für einen Empfängerkanal mit dem Namen TO.QM1 in einem WS-Manager mit dem Namen QM1 ausgewählt. ECDHE_ECDSA_AES_128_CBC_SHA256 ist eine CipherSpec, sodass QM1 ein persönliches Zertifikat mit einem Elliptic Curve-Schlüssel und einer ECDSA-basierten digitalen Signatur haben muss. Alle Clients und anderen Warteschlangenmanager, die direkt mit QM1 kommunizieren, müssen daher über digitale Zertifikate verfügen, die die CipherSpec des Typs 1 erfüllen. Andere Kanäle, die eine Verbindung zum Warteschlangenmanager QM1 herstellen, können andere CipherSpecs verwenden (z. B. ECDHE_ECDSA_3DES_EDE_CBC_SHA256), aber sie können nur Typ 1 CipherSpecs für die Kommunikation mit QM1verwenden.

Berücksichtigen Sie bei der Planung Ihrer WebSphere MQ -Netze sorgfältig, welche Kanäle SSL oder TLS erfordern, und stellen Sie sicher, dass alle Clients und Warteschlangenmanager, die interagieren müssen, denselben Typ von CipherSpecs und geeignete digitale Zertifikate verwenden. Die IETF-Standards RFC 4492, RFC 5246 und RFC 6460 beschreiben die detaillierte Verwendung von Elliptic Curve CipherSpecs in TLS 1.2.

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.

Die Ausführung des Befehls **runmqakm** erzeugt eine Ausgabe, in der der Typ des öffentlichen Schlüssels angezeigt wird:

```
Label : ibmwebspheremqexample
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 das Tool **iKeycmd (runmqckm)** mit denselben Parametern verwenden. Außerdem kann die **iKeyman (strmqikm)** -GUI verwendet werden, um Algorithmen für digitale Signaturen anzuzeigen, wenn Sie das Schlüsselrepository öffnen und doppelt auf die Bezeichnung des Zertifikats klicken. Es wird jedoch empfohlen, das Tool **runmqakm** zum Anzeigen digitaler Zertifikate zu verwenden, da es eine größere Bandbreite von Algorithmen unterstützt.

Elliptic Curve CipherSpecs und NSA Suite B

Wenn WebSphere MQ für die Konformität mit dem Suite B-konformen TLS 1.2 -Profil konfiguriert ist, sind die zulässigen CipherSpecs und digitalen Signaturalgorithmen wie in „NSA Suite B-Verschlüsselung in IBM WebSphere MQ” auf Seite 32beschrieben 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 3. Unterstützte NIST-Elliptische Kurven

NIST FIPS 186-3-Kurvenname	RFC 4492-Kurvenname	Elliptische Kurvenschlüsselgröße (Bit)
P-256	secp256r1	256
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 angeben” auf Seite 230

Geben Sie eine CipherSpec an, indem Sie den Parameter **SSLCIPH** im MQSC-Befehl **DEFINE CHANNEL** oder im MQSC-Befehl **ALTER CHANNEL** verwenden.

„Angeben, dass nur FIPS-zertifizierte CipherSpecs während der Ausführung auf dem MQI-Client verwendet werden" auf Seite 115
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 WebSphere MQ" auf Seite 32
Dieser Abschnitt enthält Informationen zur Konfiguration von IBM WebSphere MQ auf Windows-, Linux- und UNIX-Systemen für die Konformität mit dem Suite B-konformen TLS 1.2 -Profil.

„National Security Agency (NSA) Suite B Cryptography" auf Seite 21
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.

In IBM WebSphere MQ unterstützte CipherSpec -Werte

Die Gruppe der Standard-CipherSpecs lässt nur die folgenden Werte zu:

TLS 1.0

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

TLS 1.2

- ECDHE_ECDSA_AES_128_CBC_SHA256
- ECDHE_ECDSA_AES_256_CBC_SHA384
- ECDHE_ECDSA_AES_128_GCM_SHA256
- ECDHE_ECDSA_AES_256_GCM_SHA384
- ECDHE_RSA_AES_128_CBC_SHA256
- ECDHE_RSA_AES_256_CBC_SHA384
- ECDHE_RSA_AES_128_GCM_SHA256
- ECDHE_RSA_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384

Veraltete CipherSpecs aktivieren

Standardmäßig ist es Ihnen nicht erlaubt, eine veraltete CipherSpec in einer Kanaldefinition anzugeben. Wenn Sie versuchen, eine veraltete CipherSpecanzugeben, erhalten Sie die Nachricht AMQ9788 im Fehlerprotokoll für den Warteschlangenmanager.

Sie können die veralteten CipherSpecs wieder aktivieren, indem Sie die Datei `qm.ini` bearbeiten. Fügen Sie in der SSL-Zeilengruppe der Datei `qm.ini` die folgende Zeile hinzu:

```
SSL:
AllowWeakCipherSpec=Yes
```

Sie können auch eine oder mehrere der veralteten CipherSpecs zur Laufzeit auf dem Server wieder aktivieren, indem Sie die Umgebungsvariable *AMQ_SSL_WEAK_CIPHER_ENABLE* auf einen beliebigen Wert setzen. Diese Umgebungsvariable aktiviert die CipherSpecs unabhängig von dem in der Datei `qm.ini` angegebenen Wert.

Kanalauthentifizierungsdatensätze

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 WebSphere 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 4. 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 Verbindungsversuche von IP-Adressen aus geben, die keinen Zugriff auf Ihr WebSphere MQ-System haben sollten und vorübergehend blockiert werden müssen, und zwar noch bevor die Firewall aktualisiert werden kann. Diese Verbindungsversuche stammen möglicherweise nicht einmal von WebSphere MQ -Kanälen, sondern von anderen Socketanwendungen, die fehlerhaft für Ihr WebSphere MQ -Empfangsprogramm konfiguriert sind. 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 193.

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 194.

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 „Zugriff für eine vom Client bestätigte Benutzer-ID blockieren" auf Seite 197.

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 197.

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 „Zugriff durch einen definierten SSL-Namen blockieren" auf Seite 198.

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 198.

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 195.

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

Wenn festgelegt werden soll, dass bei einer Verbindung von einem WebSphere MQ-Client aus, bei der eine bestimmte Benutzer-ID verwendet wird, ein anderer, vorgegebener MCAUSER-Wert verwendet werden

soll, müssen Sie einen Kanalauthentifizierungdatensatz des Typs USERMAP setzen. Bei der Zuordnung von Benutzer-IDs sind Platzhalterzeichen nicht zulässig.

Ein Beispiel finden Sie in „Bestätigte Client-Benutzer-ID einer MCAUSER-Benutzer-ID zuordnen” auf Seite 196 .

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 196.

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 195 .

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 Kanalauthentifizierungsdatensatz 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 Kanalauthentifizierungssatz mit einem SSL- oder TLS-DN hat Vorrang vor einem Satz mit einer Benutzer-ID, einem Warteschlangenmanager-Namen oder einer IP-Adresse.
  - 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.

Stimmen mehrere Kanalauthentifizierungsdatensätze mit einem Kanalnamen, einer IP-Adresse, einem Warteschlangenmanagernamen oder einem SSL- oder TLS-DN überein, wird der Datensatz verwendet, der die höchste Übereinstimmung aufweist. Dieser Datensatz mit der höchsten Übereinstimmung wird wie folgt ermittelt:

- Für einen Kanalnamen:
  - Die genaueste Übereinstimmung erhält man mit einem Namen ohne Platzhalterzeichen; Beispiel: A.B.C.
  - Die allgemeinste Übereinstimmung erhält man mit einem einzelnen Stern (*), mit dem alle Warteschlangenmanagernamen erfasst werden.
  - Ein Muster mit einem Stern am Anfang ist allgemeiner als ein Muster mit einer definierten Zeichenfolge am Anfang, d. h., *.B.C ist ein allgemeineres Muster als A.*.
  - Ein Muster mit einem Stern an zweiter Stelle ist allgemeiner als ein Muster mit einer spezifischen Zeichenfolge an zweiter Stelle; dasselbe gilt für alle weiteren Stellen. Daher ist A.*.C ein allgemeineres Muster als A.B.*
  - Haben zwei oder mehr Muster einen Stern an der gleichen Stelle, ist das Muster mit weniger Knoten hinter dem Stern allgemeiner. So A.* allgemeiner als A.*.C.
- Für eine IP-Adresse:
  - Die genaueste Übereinstimmung erhält man mit einem Namen ohne Platzhalterzeichen; Beispiel: 192.0.2.6.
  - Die allgemeinste Übereinstimmung erhält man mit einem einzelnen Stern (*), mit dem alle Warteschlangenmanagernamen erfasst werden.
  - Ein Muster mit einem Stern am Anfang ist allgemeiner als ein Muster mit einer definierten Zeichenfolge am Anfang, d. h., *.0.2.6 ist allgemeiner als 192.*.
  - Ein Muster mit einem Stern an zweiter Stelle ist allgemeiner als ein Muster mit einer spezifischen Zeichenfolge an zweiter Stelle; dasselbe gilt für alle weiteren Stellen. Daher ist 192.*.2.6 allgemeiner als 192.0.*.
  - Haben zwei oder mehr Muster einen Stern an der gleichen Stelle, ist das Muster mit weniger Knoten hinter dem Stern allgemeiner. 192 .* allgemeiner als 192.*.2.*.
  - 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 kein doppeltes Semikolon zusammen mit einem Stern enthalten, da damit keine eindeutige Adresse gefunden werden kann. So kann 2001::* beispielsweise 2001:0000:*, 2001:0000:0000:* usw. darstellen.
- Für einen Warteschlangenmanagernamen:
  - Die genaueste Übereinstimmung erhält man mit einem Namen ohne Platzhalterzeichen; Beispiel: 192.0.2.6.
  - Die allgemeinste Übereinstimmung erhält man mit einem einzelnen Stern (*), mit dem alle Warteschlangenmanagernamen erfasst werden.
  - Ein Muster mit einem Stern am Anfang ist allgemeiner als ein Muster mit einer definierten Zeichenfolge am Anfang, d. h., *QUEUEMANAGER ist allgemeiner als QUEUEMANAGER*.
  - Ein Muster mit einem Stern an zweiter Stelle ist allgemeiner als ein Muster mit einer spezifischen Zeichenfolge an zweiter Stelle; dasselbe gilt für alle weiteren Stellen. d. h., Q*MANAGER ist allgemeiner als QUEUE*.
  - Haben zwei oder mehr Muster einen Stern an der gleichen Stelle, ist das Muster mit weniger Zeichen hinter dem Stern allgemeiner, d. h., Q* ist allgemeiner als Q*MGR.
- Bei einem SSL- oder TLS-DN gilt für die DN-Unterzeichenfolgen die folgende Reihenfolge:

Tabelle 5. 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
5	CN=	Allgemeiner Name
6	T =	Titel
7	OU=	Organisationseinheit
8	DC=	Domänenkomponente
9	O=	Organization
10	STREET=	Straße / Erste Adresszeile
11	L=	Standort
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 WebSphere MQ dem Kanalauthentifizierungsdatensatz für O=IBM den Vorzug vor dem für C=K.

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 für die oberste OU ein Platzhalterzeichen angegeben ist (OU=*), wird der DN mit den meisten OU-Attributen als spezifischer angesehen.
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

Um Kanalauthentifizierungsdatensätze anzuzeigen, verwenden Sie den MQSC-Befehl **DISPLAY CHLAUTH** oder den PCF-Befehl **Inquire Channel Authentication Records**. 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 59
Dieser Abschnitt befasst sich mit Aspekten der Sicherheit im fernen Messaging.

Nachrichtensicherheit in IBM WebSphere MQ

Nachrichtensicherheit in der IBM WebSphere MQ -Infrastruktur wird von einer separat lizenzierten Komponente bereitgestellt IBM WebSphere MQ Advanced Message Security.

IBM WebSphere MQ Advanced Message Security (AMS) erweitert die IBM WebSphere MQ -Sicherheitsservices, um Datensignierung und Verschlüsselung 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

„IBM WebSphere MQ Advanced Message Security” auf Seite 285

IBM WebSphere MQ Advanced Message Security (AMS) ist eine separat lizenzierte Komponente von IBM WebSphere MQ Advanced Message Security , die ein hohes Maß an Schutz für sensible Daten bietet, die durch das IBM WebSphere MQ Advanced Message Security -Netz fließen, ohne die Endanwendungen zu beeinträchtigen.

Sicherheitsanforderungen planen

In dieser Themensammlung finden Sie Informationen zu den Aspekten, die Sie bei der Planung der Sicherheit in einer IBM WebSphere MQ-Umgebung berücksichtigen müssen.

Sie können IBM WebSphere 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.

WebSphere MQ bietet eine Reihe von Sicherheitsservices auf Verbindungsebene, einschließlich Unterstützung für Secure Sockets Layer (SSL) und Transport Layer Security (TLS).

Bei der Implementierung von WebSphereMQ müssen Sie bestimmte Aspekte der Sicherheit berücksichtigen. Wenn Sie auf UNIX-, Linux -und Windows -Systemen diese Aspekte ignorieren und nichts tun, können Sie WebSphere MQnicht verwenden.

Sicherheitsaspekte werden unten beschrieben.

Berechtigung zur Verwaltung von WebSphere MQ

WebSphere MQ -Administratoren benötigen die folgende Berechtigung:

- Befehle für die Verwaltung von WebSphere MQ ausgeben
- IBM WebSphere MQ Explorer verwenden

Weitere Informationen finden Sie unter:

- „Berechtigung zur Verwaltung von IBM WebSphere MQ auf UNIX, Linux, and Windows -Systemen” auf Seite 208

Berechtigung zum Arbeiten mit WebSphere MQ -Objekten

Anwendungen können über MQI-Aufrufe auf die folgenden WebSphere MQ -Objekte zugreifen:

- Warteschlangenmanager
- Warteschlangen
- Prozesse
- Namenslisten
- Themen

Anwendungen können auch PCF-Befehle (Programmable Command Format) verwenden, um auf diese WebSphere MQ -Objekte sowie auf Kanäle und Authentifizierungsinformationsobjekte zuzugreifen. Diese Objekte können durch WebSphere MQ geschützt werden, sodass die Benutzer-IDs, die den Anwendungen zugeordnet sind, die Berechtigung für den Zugriff auf sie benötigen.

Weitere Informationen finden Sie unter „Berechtigung für Anwendungen zur Verwendung von IBM WebSphere MQ” auf Seite 53.

Kanalsicherheit

Die Benutzer-IDs, die Nachrichtenkanalagenten (MCAs) zugeordnet sind, benötigen die Berechtigung zum Zugriff auf verschiedene WebSphere 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 unter „Kanalberechtigung” auf Seite 74.

Weitere Überlegungen

Sie müssen die folgenden Aspekte der Sicherheit nur berücksichtigen, wenn Sie bestimmte WebSphere MQ -Funktionen oder Basisprodukterweiterungen verwenden:

- „Sicherheit für WS-Manager-Cluster” auf Seite 83
- „Sicherheit für IBM WebSphere MQ Publish/Subscribe” auf Seite 84
- „Sicherheit für IBM WebSphere MQ Internet Pass-thru” auf Seite 86

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 WebSphere 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 WebSphere MQ-Kanäle, die SSL oder TLS verwenden, nutzen digitale Zertifikate als Verfahren zur Identifikation und Authentifizierung. Jedes digitale Zertifikat verfügt über einen registrierten Namen, der anhand von Kanalauthentifizierungsdatensätzen auf bestimmte Identitäten abgebildet werden kann. Darüber hinaus legen CA-Zertifikate im Schlüsselrepository fest, welche digitalen Zertifikate für die Authentifizierung bei IBM WebSphere MQ verwendet werden können. Weitere Informationen finden Sie unter:

- „Zuordnung eines fernen Warteschlangenmanagers zu einer MCAUSER-Benutzer-ID” auf Seite 195
- „Bestätigte Client-Benutzer-ID einer MCAUSER-Benutzer-ID zuordnen” auf Seite 196
- „Zuordnen eines SSL-oder TLS-definierten Namens zu einer MCAUSER-Benutzer-ID” auf Seite 196
- „Zuordnen einer IP-Adresse zu einer MCAUSER-Benutzer-ID” auf Seite 198

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 WebSphere MQ MQI-Client, der mit dem Server verbunden ist. Die Sicherheit wird auf vier Ebenen angewendet, wie im folgenden Text beschrieben. MCA ist ein Nachrichtenkanalagent.

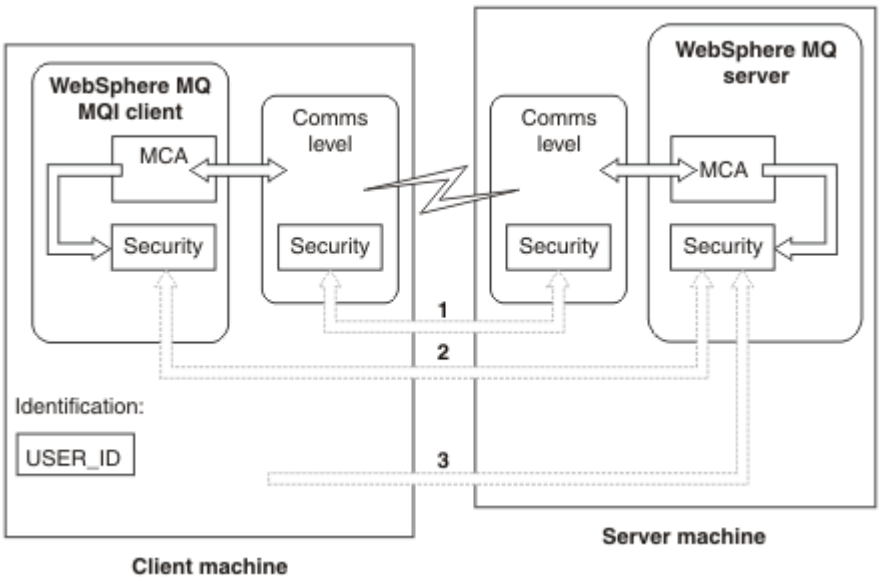


Abbildung 7. Sicherheit in einer Client/Server-Verbindung

1. Übertragungsstufe

   Siehe Pfeil 1. Verwenden Sie SSL oder TLS, um die Sicherheit auf Kommunikationsebene zu implementieren. Weitere Informationen finden Sie in „Verschlüsselte Sicherheitsprotokolle: SSL und TLS” auf Seite 15.

2. Kanalauthentifizierungsdatensätze

   Siehe Pfeile 2 & 3. Die Authentifizierung kann über die IP-Adresse oder über definierte SSL/TLS-Namen auf Sicherheitsebene 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 „Kanalauthentifizierungsdatensätze” auf Seite 42.

3. 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 .

4. 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 WebSphere 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.

   Windows-Clients senden außerdem zusätzliche Informationen zur Unterstützung der Identifikation.

   - 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.

   Zur Unterstützung der Identifikation auf IBM WebSphere MQ Client for HP Integrity NonStop Server übergibt der Client den OSS-Sicherheitsalias, unter dem die Clientanwendung ausgeführt wird. Diese ID hat in der Regel das Format <PRIMARYGROUP>.<ALIAS>. Falls erforderlich, können Sie diese Benutzer-ID mithilfe von Kanalauthentifizierungsdatensätzen oder einem Sicherheitsexit einer alternativen Benutzer-ID auf dem Warteschlangenmanager zuordnen. Weitere Informationen zu Nachrichtenexits finden Sie unter „Identitätsabgleich in Nachrichtenexits” auf Seite 158. Weitere Informationen

zum Definieren von Kanalauthentifizierungsdatensätzen finden Sie unter „Bestätigte Client-Benutzer-ID einer MCAUSER-Benutzer-ID zuordnen” auf Seite 196.

Die Werte der Benutzer-ID und, falls verfügbar, die Sicherheits-ID können vom Serversicherheitsexit verwendet werden, um die Identität des IBM WebSphere MQ MQI-Clients zu ermitteln.

Benutzer-IDs

Wenn sich der IBM WebSphere MQ MQI-Client unter Windows und der IBM WebSphere MQ -Server auch unter Windows befindet und Zugriff auf die Domäne hat, in der die Client-Benutzer-ID definiert ist, unterstützt IBM WebSphere MQ Benutzer-IDs mit bis zu 20 Zeichen. Auf UNIX and Linux -Plattformen und -Konfigurationen beträgt die maximale Länge 12 Zeichen.

Ein WebSphere MQ for Fenster -Server unterstützt die Verbindung eines Fenster -Clients nicht, wenn der Client unter einer Benutzer-ID ausgeführt wird, die das Zeichen @ enthält, z. B. `abc@d`. Der Rückkehrcode des `MQCONN` -Aufrufs 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`.

Beachten Sie, dass `UNKNOWN` eine reservierte Benutzer-ID ist und die `NOBODY` -Benutzer-ID auch eine besondere Bedeutung für WebSphere MQhat. Das Erstellen von Benutzer-IDs im Betriebssystem mit dem Namen `UNKNOWN` oder `NOBODY` kann unbeabsichtigte Ergebnisse haben.

Obwohl Benutzer-IDs für die Authentifizierung verwendet werden, werden Gruppen für die Berechtigung verwendet, mit Ausnahme von Windows.

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.

Planungsberechtigung

Planen Sie die Benutzer, die über Administratorberechtigung verfügen, und planen Sie, wie Benutzer von Anwendungen berechtigt werden, IBM WebSphere MQ -Objekte ordnungsgemäß zu verwenden, einschließlich derer, die über einen IBM WebSphere MQ MQI-Client eine Verbindung herstellen.

Einzelpersonen oder Anwendungen müssen Zugriffsberechtigungen erteilt werden, damit IBM WebSphere 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 WebSphere MQ kann in zwei Hauptkategorien unterteilt werden:

- Berechtigung zum Ausführen von Verwaltungsoperationen
- Berechtigung für Anwendungen zur Verwendung von IBM WebSphere 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 zur Verwaltung von IBM WebSphere MQ

IBM WebSphere MQ-Administratoren benötigen die Berechtigung zum Ausführen verschiedener Funktionen. Diese Berechtigung wird auf unterschiedliche Weise auf verschiedenen Plattformen abgerufen.

IBM WebSphere MQ -Administratoren benötigen die Berechtigung für:

- Befehle zur Verwaltung von IBM WebSphere MQ ausgeben
- Verwenden Sie bitteIBM WebSphere MQ Explorer

Weitere Informationen finden Sie im entsprechenden Thema zu Ihrem Betriebssystem.

Berechtigung zur Verwaltung von IBM WebSphere MQ auf UNIX -und Windows -Systemen

Ein IBM WebSphere MQ-Administrator ist ein Mitglied der Gruppe 'mqm'. Diese Gruppe verfügt über Zugriff auf alle IBM WebSphere MQ -Ressourcen und kann IBM WebSphere MQ -Steuerbefehle ausgeben. Ein Administrator kann anderen Benutzern bestimmte Berechtigungen erteilen.

Als IBM WebSphere MQ -Administrator auf UNIX -und Windows -Systemen muss ein Benutzer zur Gruppe *mqm*gehören. Diese Gruppe wird automatisch erstellt, wenn Sie WebSphere MQinstallieren. Um Benutzern die Ausgabe von Steuerbefehlen zu ermöglichen, müssen Sie sie zur Gruppe 'mqm' hinzufügen. Dies schließt den Rootbenutzer auf UNIX -Systemen ein.

Benutzer, die nicht Mitglied der Gruppe mqm sind, können Verwaltungsberechtigungen erteilen, sie können jedoch keine IBM WebSphere MQ -Steuerbefehle ausgeben, und sie sind berechtigt, nur die Befehle auszuführen, für die ihnen Zugriff erteilt wurde.

Außerdem haben die Konten SYSTEM und Administrator auf Windows -Systemen uneingeschränkten Zugriff auf IBM WebSphere MQ -Ressourcen.

Alle Mitglieder der Gruppe 'mqm' haben Zugriff auf alle WebSphere MQ -Ressourcen auf dem System, einschließlich der Möglichkeit, jeden Warteschlangenmanager auf dem System zu verwalten. Dieser Zugriff kann nur widerrufen werden, wenn ein Benutzer aus der Gruppe 'mqm' entfernt wird. Auf Windows -Systemen haben Mitglieder der Gruppe 'Administratoren' auch Zugriff auf alle WebSphere MQ -Ressourcen.

Administratoren können den Steuerbefehl **runmqsc** verwenden, um MQSC-Befehle (WebSphere MQ Script) 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 WebSphere MQ Explorer gibt PCF-Befehle aus, um Verwaltungstasks auszuführen. Administratoren benötigen keine zusätzlichen Berechtigungen für die Verwendung von WebSphere MQ Explorer, um einen WS-Manager auf dem lokalen System zu verwalten. Wenn der WebSphere MQ Explorer zum Verwalten eines Warteschlangenmanagers auf einem anderen System verwendet wird, müssen Administratoren über die erforderlichen Berechtigungen für die PCF-Befehle verfügen, die vom fernen Warteschlangenmanager verarbeitet werden sollen.

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, die auf Warteschlangenmanagern, Warteschlangen, Kanälen, Prozessen, Namenslisten und Authentifizierungsinformationsobjekten ausgeführt werden, finden Sie in „Berechtigung für Anwendungen zur Verwendung von IBM WebSphere MQ” auf Seite 53.
- Informationen zu Befehlen, die auf Kanälen, Kanalinitiatoren, Empfangsprogrammen und Clustern ausgeführt werden, finden Sie unter Kanalsicherheit .

Weitere Informationen zu der Berechtigung, die Sie zur Verwaltung von WebSphere MQ auf UNIX -und Windows -Systemen benötigen, finden Sie in den Referenzinformationen.

Berechtigung für Anwendungen zur Verwendung von IBM WebSphere 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 WebSphere MQ-Objekte zugreifen:

- Warteschlangenmanager
- Warteschlangen
- Prozesse
- Namenslisten
- Themen

Anwendungen können auch PCF-Befehle verwenden, um IBM WebSphere MQ-Objekte zu verwalten. Wenn der PCF-Befehl verarbeitet wird, verwendet er den Berechtigungskontext der Benutzer-ID, die die PCF-Nachricht eingibt.

Anwendungen umfassen in diesem Kontext die von Benutzern und Anbietern geschriebenen Anwendungen.

Anwendungen, die IBM WebSphere MQ -Klassen für Java, IBM WebSphere MQ -Klassen für JMS, IBM WebSphere MQ -Klassen für .NET oder Message Service Clients for C/C++ and .NET verwenden, verwenden die MQI indirekt.

MCAs geben auch MQI-Aufrufe aus und die Benutzer-IDs, die den MCAs zugeordnet sind, benötigen die Berechtigung zum Zugriff auf diese WebSphere MQ -Objekte. Weitere Informationen zu diesen Benutzer-IDs und den erforderlichen Berechtigungen finden Sie in „Kanalberechtigung” auf Seite 74.

Wenn Berechtigungsprüfungen durchgeführt werden

Berechtigungsprüfungen werden durchgeführt, wenn eine Anwendung versucht, auf einen WS-Manager, eine Warteschlange, einen Prozess oder eine Namensliste zuzugreifen.

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 WebSphere MQ anmelden. IBM WebSphere MQ setzt voraus, dass sich die Benutzer am zugrunde liegenden Betriebssystem angemeldet haben und dort authentifiziert sind.

Wenn eine Anwendung ein IBM WebSphere 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 WS-Manager ausgeführt. Unter UNIX, Linux, and Windowswird 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. Die Berechtigungsprüfungen können Prüfungen auf mehr als ein Themenobjekt beinhalten.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, der für ein WebSphere MQ -Objekt ausgeführt wird, 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 56.

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 festlegen 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
Der Typ der Anwendung, die die Nachricht eingereiht hat, 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 .

Berechtigung zum Arbeiten mit IBM WebSphere MQ -Objekten auf UNIX-, Linux -und Windows -Systemen

Die Berechtigungsservicekomponente, die mit IBM WebSphere MQ bereitgestellt wird, wird als *Objektberechtigungsmanager (OAM)* bezeichnet. Sie ermöglicht die Zugriffssteuerung über Authentifizierungs-und Berechtigungsprüfungen.

1. Authentifizierung.

   Die Authentifizierungsprüfung, die von dem mit IBM WebSphere 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.

   Der OAM führt seine Authentifizierungs-Prüfung durch, wenn eine Anwendung eine Verbindung zu einem Warteschlangenmanager herstellt, und die folgenden Bedingungen erfüllt sind.

   Wenn eine MQCSP-Struktur von der Verbindungsanwendung bereitgestellt wurde und das Attribut *AuthenticationType* in der MQCSP-Struktur den Wert MQCSP_AUTH_USER_ID_AND_PWD erhält, wird die Prüfung durch den OAM in seiner Funktion MQZID_AUTHENTICATE_USER ausgeführt. Dies ist die Prüfung: Die Benutzer-ID in der MQCSP-Struktur wird mit der Benutzer-ID in der *IdentityContext* (MQZIC) verglichen, um festzustellen, ob sie übereinstimmen. Wenn sie nicht übereinstimmen, schlägt die Prüfung fehl.

   Diese Basisprüfung soll keine vollständige Authentifizierung des Benutzers sein. Es gibt beispielsweise keine Überprüfung der Authentizität des Benutzers durch Überprüfung des in der MQCSP-Struktur

angegebenen Kennworts. Wenn die Anwendung eine MQCSP-Struktur ausnimmt, wird auch keine Prüfung durchgeführt.

Der mit IBM WebSphere MQ bereitgestellte OAM bietet keine vollständigeren Authentifizierungsservices an, die möglicherweise im Warteschlangenmanager über die Berechtigungsservicekomponente erforderlich sind. Sie müssen eine neue Berechtigungsservicekomponente schreiben oder einen von einem Lieferanten anfordern.

2. 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 UNIX-, Linux- und Windows-Systemen übernimmt der *Berechtigungsservice* die Zugriffssteuerung, wenn eine Anwendung einen MQI-Aufruf für den Zugriff auf ein IBM WebSphere MQ-Objekt ausgibt, sofern es sich bei dem Objekt um einen Warteschlangenmanager, eine Warteschlange, einen Prozess, ein Topic oder eine Namensliste handelt. Dazu gehören Prüfungen auf alternative Benutzerberechtigung und die Berechtigung zum Festlegen oder Übergeben von Kontextinformationen.

Unter Windows erteilt der OAM den Mitgliedern der Administratorgruppe die Berechtigung, auf alle IBM WebSphere MQ-Objekte zuzugreifen, selbst wenn UAC aktiviert ist.

Außerdem hat das Konto SYSTEM auf Windows -Systemen uneingeschränkten Zugriff auf IBM WebSphere MQ -Ressourcen.

Der Berechtigungsservice stellt zusätzlich Berechtigungsprüfungen bereit, wenn ein PCF-Befehl eines dieser IBM WebSphere MQ-Objekte oder ein Authentifizierungsdatenobjekt ausführt. Der entsprechende MQSC-Befehl, der in einem Escape-PCF-Befehl eingebunden ist, wird auf die gleiche Weise behandelt.

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 WebSphere MQ-Produkten bereitgestellten Komponenten erweitert oder ersetzt werden.

Die Berechtigungsservicekomponente, die mit IBM WebSphere MQ bereitgestellt wird, wird als *Objektberechtigungsmanager (OAM)* bezeichnet. Der OAM wird automatisch für jeden Warteschlangenmanager, den Sie erstellen, aktiviert.

Der OAM verwaltet eine Zugriffssteuerungsliste (ACL) für jedes IBM WebSphere MQ -Objekt, auf das der Zugriff gesteuert wird. Auf Systemen mit UNIX and Linux können nur Gruppen-IDs in einer ACL angezeigt werden. Dies bedeutet, dass alle Mitglieder einer Gruppe die gleichen Berechtigungen haben. Auf Windows -Systemen können sowohl Benutzer-IDs als auch 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.AIX und Linux haben diesen Grenzwert erhöht, aber IBM WebSphere MQ unterliegt weiterhin einer Beschränkung von 12 Zeichen auf allen UNIX -Plattformen. Wenn Sie eine Benutzer-ID mit mehr als 12 Zeichen verwenden, ersetzt IBM WebSphere 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 WebSphere 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 .

Auf Systemen mit UNIX, Linux und Windows erteilt und widerruft der Steuerbefehl **setmqaut** Berechtigungen und wird zur Verwaltung der ACLs verwendet. 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 zu stellen, deren Name 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.

Wenn Sie keine Berechtigungsprüfungen wünschen, z. B. in einer Testumgebung, können Sie den OAM inaktivieren.

PCF für den Zugriff auf OAM-Befehle verwenden
Auf UNIX-, Linux -und Windows -Systemen können Sie mithilfe von PCF-Befehlen auf OAM-Verwaltungsbefehle zugreifen.

Die PCF-Befehle und die entsprechenden OAM-Befehle lauten wie folgt:

Tabelle 6. 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 enthält Einführung in Programmable Command Formats .

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 WebSphere MQ -Funktionen zur Verfügung stellen. 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 einen MCAUSER-Wert auf Basis der folgenden Informationen festzulegen: ferne IP-Adresse, ferne Benutzer-ID, angegebener definierter Name des SSL- oder TLS-Subjekts oder Name 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.

Sicherheit von Objekten auf UNIX and Linux -Systemen

Verwaltungsbenutzer müssen Mitglieder der Gruppe 'mqm' auf Ihrem System sein (einschließlich Root), wenn mit dieser ID die Verwaltungsbefehle von IBM WebSphere MQ verwendet werden sollen.

Sie sollten amqcrsta immer als die Benutzer-ID "mqm" ausführen.

Benutzer-IDs auf UNIX and Linux -Systemen

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.

Sicherheit von Objekten auf Windows -Systemen

Verwaltungsbenutzer müssen sowohl zur Gruppe 'mqm' als auch zur Administratorgruppe auf Windows -Systemen gehören, wenn diese ID IBM WebSphere MQ -Verwaltungsbefehle verwenden soll.

Benutzer-IDs auf Windows -Systemen

Wenn auf Windows -Systemen *kein Nachrichtenexit installiert ist*, konvertiert der Warteschlangenmanager 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.

Benutzer-IDs auf mehreren Systemen

Auf anderen Plattformen als Windows verwenden UNIX and Linux -Systeme Großbuchstaben für Benutzer-IDs in Nachrichten.

Damit Windows-und UNIX and Linux -Systeme Benutzer-IDs in Kleinbuchstaben in Nachrichten verwenden können, werden die folgenden Konvertierungen vom 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 automatischen Konvertierungen werden nicht ausgeführt, wenn Sie einen Nachrichtenexit auf UNIX-, Linux -und Windows -Systemen aus einem anderen Grund bereitstellen.

Angepasster Berechtigungsservice verwenden

IBM WebSphere MQ stellt einen installierbaren Berechtigungsservice bereit. Sie können auswählen, dass ein alternativer Service installiert werden soll.

Die mit IBM WebSphere MQ gelieferte 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 WebSphere 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 7. 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 62 .
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 verwendete ID auf andere Weise festgelegt wird.

Tabelle 7. Die Benutzer-ID, die von einem Serververbindungskanal verwendet wird. (Forts.)	
Benutzer-ID	**Bei Verwendung**
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 63 .

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

Bei IBM WebSphere MQ MQI-Clients ist der Prozess, der die MQI-Aufrufe ausgibt, der MCA der 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.

Stellen Sie sicher, dass das Feld MCAUSER auf Windows-Plattformen auf 12 Zeichen beschränkt ist, weil zusätzliche Zeichen abgeschnitten werden, was zu Berechtigungsfehlern führen kann.

Der IBM WebSphere 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:

- 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 (nichtz/OS) die Benutzer-ID aus dem Eintrag `inetd.conf` oder die Benutzer-ID, mit der der Listener gestartet wurde.
- Für SNA (nichtz/OS) die Benutzer-ID aus dem SNA-Servereintrag oder (falls keine vorhanden ist) die eingehende Verbindungsanforderung oder die Benutzer-ID, die das Empfangsprogramm gestartet hat.
- 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 WebSphere 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 WebSphere 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.

Für Server auf UNIX-, Linux- und Windows-Systemen wird der Inhalt des vom Client empfangenen Felds `MCAUserIdentifier` in Kleinbuchstaben umgesetzt.

Für Server unter IBM i werden die Inhalte des Felds `LongMCAUserIdentifier`, das vom Client empfangen wird, in Großbuchstaben geändert.

Für Server auf UNIX and Linux-Systemen werden die Inhalte des Felds `LongMCAUserIdentifier`, das vom Client empfangen wird, in Kleinbuchstaben geändert.

Bei der Benutzer-ID, die bei der Verwendung einer Anwendung mit einer MQ JMS-Bindung übergeben wird, handelt es sich standardmäßig um 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 SSL oder TLS verwenden, in diesem Fall müssen Sie Ihre Nutzung von digitalen Zertifikaten 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 64
Dieses Thema enthält Informationen zu verschiedenen Aspekten der Sicherheit auf Verbindungsebene und Sicherheit auf Anwendungsebene und vergleicht die beiden Sicherheitsstufen.

„Kanalexitprogramme” auf Seite 69

Kanalexitprogramme sind Programme, die an definierten Stellen in der Verarbeitungsreihenfolge eines MCA aufgerufen werden. Benutzer und Anbieter können ihre eigenen Kanalexitprogramme schreiben. Einige werden von IBMbereitgestellt.

„Kanäle mit SSL schützen” auf Seite 76
Die SSL-Unterstützung in IBM WebSphere MQ verwendet das Authentifizierungsinformationsobjekt 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 8 auf Seite 64 dargestellt.

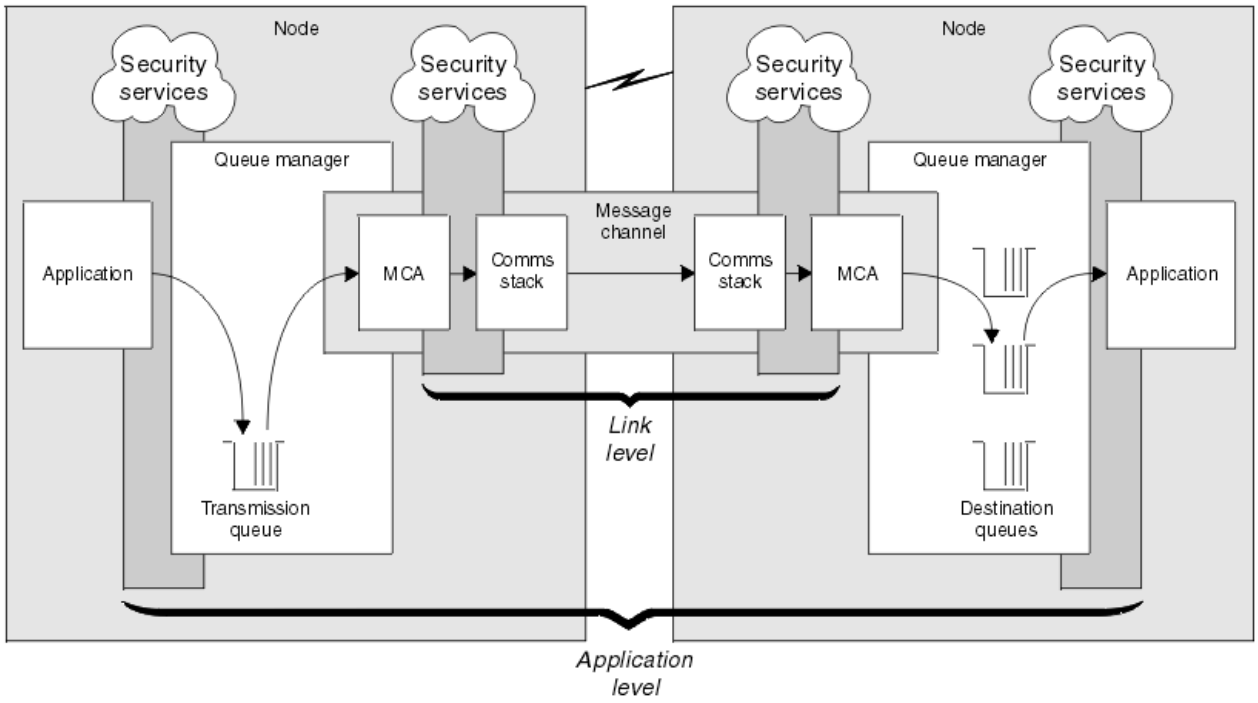


Abbildung 8. Sicherheit auf Verbindungsebene und Sicherheit auf Anwendungsebene

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.

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 8 auf Seite 64 dargestellt.

Warteschlangenmanager, die nicht in kontrollierten und gesicherten Umgebungen ausgeführt werden

Wenn ein Warteschlangenmanager in einer kontrollierten und vertrauenswürdigen Umgebung ausgeführt wird, können die von WebSphere MQ bereitgestellten Zugriffssteuerungsmechanismen als ausreichend angesehen werden, um die in seinen 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 Warteschlangenmanager 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 nicht die Daten in WebSphere MQ -Kanalprotokollflüssen schützen, die keine Nachrichtendaten sind. 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 8 auf Seite 64 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 WebSphere MQ bereitgestellte Sicherheit auf Verbindungsebene

Das primäre Mittel zur Bereitstellung von Vertraulichkeit und Datenintegrität in IBM WebSphere MQ ist die Verwendung von SSL oder TLS. Weitere Informationen zur Verwendung von SSL und TLS in IBM WebSphere MQ finden Sie unter „IBM WebSphere MQ Unterstützung für SSL und TLS” auf Seite 25. Für die Authentifizierung stellt IBM WebSphere 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 unter „Kanalauthentifizierungsdatensätze” auf Seite 42.

Sicherheit auf eigene Linkebene bereitstellen

In dieser Themensammlung wird beschrieben, wie Sie Ihre eigenen Sicherheitsservices auf Verbindungsebene bereitstellen können. Das Schreiben eigener Kanalexitprogramme ist der wichtigste Weg, um eigene Sicherheitsdienste auf Verbindungsebene bereitzustellen.

Kanalexitprogramme werden in „Kanalexitprogramme” auf Seite 69 eingeführt. In demselben Abschnitt wird auch das Kanalexitprogramm beschrieben, das mit IBM WebSphere MQ for Windows (dem SSPI-Kanalexitprogramm) bereitgestellt wird. 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 direkt oder indirekt von der Anwendung, dem Warteschlangenmanager, einem anderen Produkt, das WebSphere MQunterstützt, oder einer Kombination dieser Funktionen aufgerufen werden. Die Sicherheit auf Anwendungsebene ist in Abbildung 8 auf Seite 64 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.

Erweiterte Nachrichtensicherheit-Planung

IBM WebSphere MQ Advanced Message Security (AMS) ist eine separat lizenzierte Komponente von IBM WebSphere MQ , die ein hohes Maß an Schutz für sensible Daten bietet, die durch das IBM WebSphere MQ -Netz fließen, ohne die Endanwendungen zu beeinträchtigen.

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 WebSphere MQ entwickeln. Solche Lösungen erfordern jedoch Fachkenntnisse und können kompliziert und kostspielig sein, um sie zu erhalten. IBM WebSphere MQ 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.

IBM WebSphere MQ Advanced Message Security erweitert die Sicherheitsfunktionen von IBM WebSphere MQ auf folgende Arten:

- 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 WebSphere MQ-Server und -Clients unterstützt.

- Es integriert sich in IBM WebSphere MQ Managed File Transfer , um eine sichere Messaging-Lösung für End-to-End-Systeme bereitzustellen.

Weitere Informationen finden Sie im Abschnitt „IBM WebSphere MQ Advanced Message Security” auf Seite 285.

Bereitstellen der Sicherheit auf Anwendungsebene
In dieser Themensammlung wird beschrieben, wie Sie Ihre eigenen Sicherheitsservices auf Anwendungsebene bereitstellen können.

Damit Sie die Sicherheit auf Anwendungsebene implementieren können, stellt IBM WebSphere MQ den API-Exit und den API-Steuerübergabeexit bereit.

Diese Exits 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 WebSphere 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 WebSphere 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 von IBMbereitgestellt.

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 9 auf Seite 70 dargestellt und werden in den folgenden Abschnitten beschrieben.

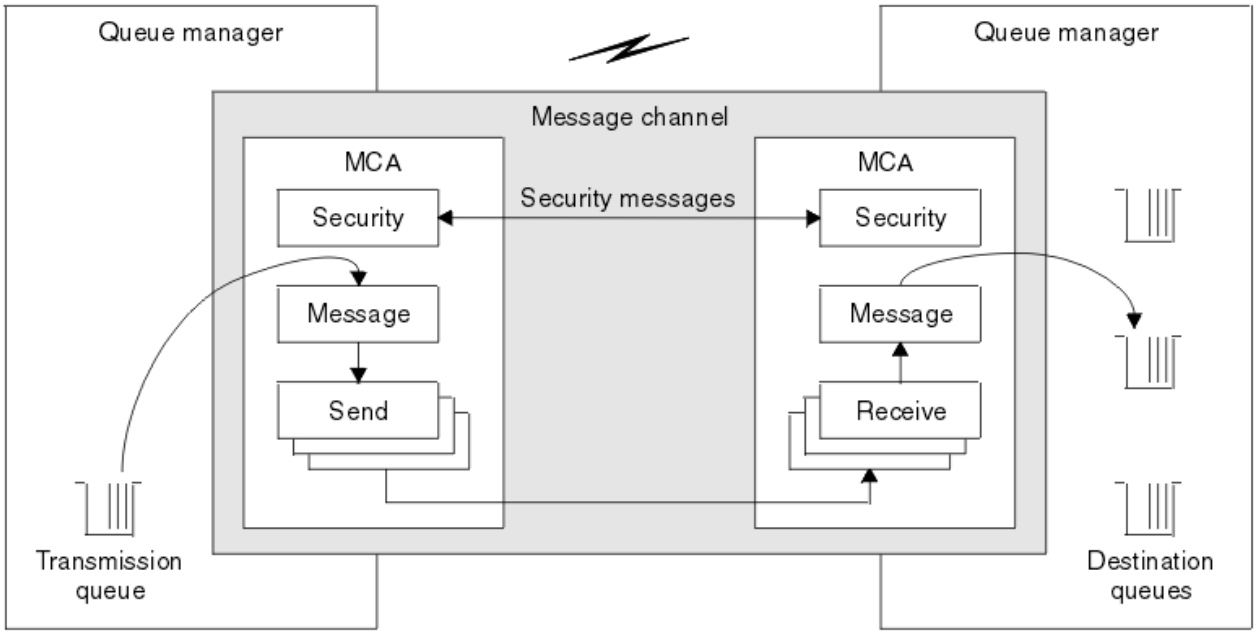


Abbildung 9. Sicherheits-, Nachrichten-, Sende-und Empfangsexits in einem Nachrichtenkanal

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 67.

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 der Zweck eines MQI-Kanals darin besteht, die Eingabe-und Ausgabeparameter von MQI-Aufrufen für den Fluss zwischen der IBM WebSphere MQ MQI-Clientanwendung und dem Warteschlangenmanager zu aktivieren.

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 67.

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 WebSphere 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 WebSphere 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 Übertragungs-

segment. 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 67.

Datenintegrität planen

Planen Sie, wie die Integrität Ihrer Daten beibehalten wird.

Sie können die Datenintegrität auf Anwendungsebene oder auf Linkebene implementieren.

Auf Anwendungsebene können Sie IBM WebSphere MQ Advanced Message Security zum digitalen Signieren von Nachrichten verwenden, um sich vor unbefugten Änderungen zu schützen. Sie können auch API-Exitprogramme verwenden, wenn Standardfunktionen Ihre Anforderungen nicht erfüllen.

Auf der Verbindungsebene können Sie SSL oder TLS verwenden. In diesem Fall müssen Sie Ihre Nutzung von digitalen Zertifikaten planen. Sie können Kanalexitprogramme auch verwenden, wenn die Standardfunktionen Ihre Anforderungen nicht erfüllen.

Zugehörige Konzepte

„Kanäle mit SSL schützen” auf Seite 76
Die SSL-Unterstützung in IBM WebSphere MQ verwendet das Authentifizierungsinformationsobjekt des Warteschlangenmanagers und verschiedene MQSC-Befehle. Sie müssen auch Ihre Verwendung digitaler Zertifikate in Betracht ziehen.

„Datenintegrität in IBM WebSphere MQ” auf Seite 23
Sie können einen Datenintegritätsservice verwenden, um festzustellen, ob eine Nachricht geändert wurde.

„ Erweiterte Nachrichtensicherheit-Planung” auf Seite 68
IBM WebSphere MQ Advanced Message Security (AMS) ist eine separat lizenzierte Komponente von IBM WebSphere MQ , die ein hohes Maß an Schutz für sensible Daten bietet, die durch das IBM WebSphere MQ -Netz fließen, ohne die Endanwendungen zu beeinträchtigen.

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 WebSphere 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 WebSphere MQ über die IBM WebSphere MQ -Schnittstellen
Konfigurieren Sie IBM WebSphere MQ für die Ausgabe von Instrumentierungsereignissen, insbesondere für Befehlsereignisse und Konfigurationsereignisse.

Änderungen an IBM WebSphere MQ außerhalb der Steuerung
Einige Änderungen können sich auf die Funktionsweise von IBM WebSphere MQ auswirken, können aber nicht direkt von IBM WebSphere 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 WebSphere MQ
Möglicherweise müssen Sie Betriebssystemtools verwenden, um Aktivitäten wie das Starten und Stoppen von Warteschlangenmanagern zu prüfen. In einigen Fällen kann IBM WebSphere MQ so konfiguriert werden, dass Instrumentierungsereignisse ausgegeben werden.

Anwendungsaktivität in IBM WebSphere MQ
Wenn Sie die Aktionen von Anwendungen überwachen möchten, z. B. das Öffnen von Warteschlangen und das Einreihen und Abrufen von Nachrichten, konfigurieren Sie IBM WebSphere MQ so, dass die entsprechenden Ereignisse ausgegeben werden.

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 WebSphere 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

Es wird eine Gruppe von Tests mit dem IBM WebSphere 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, benötigen Sie eine Benutzer-ID, die Zugriff auf verschiedene IBM WebSphere MQ -Ressourcen hat.

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 WebSphere MQkönnen Kanäle durch SSL-oder 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 Berechtigungen 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.

  Dies bezieht sich auf die Message-Queuing-Schnittstelle (MQI), sodass möglicherweise zusätzliche Zugriffssteuerungsprüfungen durchgeführt werden müssen, wenn Sie WebSphere MQ Object Authority Manager (OAM) nicht verwenden. 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 das Servicekonto des Warteschlangenmanagers, das dann über vollständige Administratorrechte verfügt und keine Sonderberechtigungen benötigt.

    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.

– Wenn Sie eine Benutzer-ID ohne WebSphere -Administratorberechtigungen verwenden, müssen Sie dieser Benutzer-ID die Berechtigung dsp und ctrlx für den Kanal erteilen, damit der Kanal funktioniert. 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. Geben Sie zum Beispiel folgende Befehle aus:

```
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
„Kanalauthentifizierungsdatensätze” auf Seite 42
Die Zugriffsberechtigungen zum Herstellen von Systemverbindungen auf Kanalebene können mithilfe von Kanalauthentifizierungsdatensätzen gezielter gesteuert werden.

Eigenschaften des Kanalauthentifizierungsdatensatzes

Kanalinitiatordefinitionen schützen
Nur Mitglieder der Gruppe mqm können Kanalinitiatoren bearbeiten.

IBM WebSphere MQ-Kanalinitiatoren sind keine IBM WebSphere MQ-Objekte; der Zugriff wird nicht vom OAM gesteuert. IBM WebSphere 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`ausgibt, muss die Benutzer-ID, die im Nachrichtendeskriptor der PCF-Nachricht angegeben ist, ein Mitglied der Gruppe mqm auf dem Zielwarteschlangenmanager sein.

Eine Benutzer-ID muss auch Mitglied der Gruppe mqm auf der Zielmaschine sein, um die entsprechenden MQSC-Befehle über den Escape-PCF-Befehl oder mit `runmqsc` im indirekten Modus auszugeben.

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 10 auf Seite 96).

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 unter „Kanalexitprogramme” auf Seite 69.

Kanäle mit SSL schützen

Die SSL-Unterstützung in IBM WebSphere MQ verwendet das Authentifizierungsinformationsobjekt des Warteschlangenmanagers und verschiedene MQSC-Befehle. Sie müssen auch Ihre Verwendung digitaler Zertifikate in Betracht ziehen.

Befehle und Attribute für SSL-Unterstützung

Das SSL-Protokoll (Secure Sockets Layer) bietet Kanalsicherheit mit Schutz vor Ausspionieren, Manipulation und Identitätswechsel. Mit der IBM WebSphere MQ -Unterstützung für SSL können Sie in der Kanaldefinition angeben, dass ein bestimmter Kanal SSL-Sicherheit verwendet. Sie können auch Details zu dem Typ der gewünschten Sicherheit angeben, z. B. den Verschlüsselungsalgorithmus, den Sie verwenden möchten.

Die folgenden MQSC-Befehle unterstützen SSL:

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 Warteschlangenmanagerparameter unterstützen SSL:

SSLCRLNL
Dieses Attribut legt eine Namensliste mit Authentifizierungsinformationsobjekten fest, durch die Zertifikatssperrstellen bereitgestellt werden, über die eine erweiterte TLS/SSL-Zertifikatsprüfung durchgeführt werden kann.

SSLCRYP
Unter Windowslegt UNIX and Linux -Systeme das Warteschlangenmanagerattribut SSLCryptoHardware fest. Dieses Attribut ist der Name der Parameterzeichenfolge, die Sie zum Konfigurieren der Verschlüsselungshardware verwenden können, die Sie auf Ihrem System haben.

SSLEV
Bestimmt, ob eine SSL-Ereignisnachricht gemeldet wird, wenn ein Kanal, der SSL verwendet, keine SSL-Verbindung herstellen kann.

SSLFIPS
Gibt an, ob nur FIPS-zertifizierte Algorithmen verwendet werden sollen, wenn die Verschlüsselung in IBM WebSphere 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 Windows-UNIX and Linux -Systemen ein Schlüsselrepository einem Warteschlangenmanager zu. Die Schlüsseldatenbank wird in einer Schlüsseldatenbank von *GSKit* gehalten. (Das IBM Global Security Kit (GSKit) ermöglicht Ihnen die Verwendung der SSL-Sicherheit unter Windows,UNIX and Linux -Systemen.)

SSLRKEYC
Die Anzahl der Bytes, die innerhalb eines SSL-Datenaustauschs gesendet und empfangen werden, bevor der geheime Schlüssel neu festgelegt wird. Die Anzahl der Byte enthält Steuerinformationen, die vom MCA gesendet wurden.

Die folgenden Kanalparameter unterstützen SSL:

SSLCAUTH
Definiert, ob IBM WebSphere MQ ein Zertifikat vom SSL-Client erfordert und validiert.

SSLCIPH
Gibt die Verschlüsselungsstärke und -funktion (CipherSpec) an, z. B. NULL_MD5 oder RC4_MD5_US. 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 Befehle `setmqaut`, `dspmqaut`, `dmpmqaut`, `rcrmqobj`, `rcdmqimg`und `dspmqfls` zur Unterstützung des Authentifizierungsinformationsobjekts beschrieben. Außerdem wird der Befehl iKeycmd für die Verwaltung von Zertifikaten auf UNIX and Linux -Systemen und das Tool 'runmqakm' für die Verwaltung von Zertifikaten auf Systemen mit UNIX, Linux und Windows beschrieben. Siehe hierzu folgenden Abschnitte:

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- Schlüssel und Zertifikate verwalten

Eine Übersicht über die Kanalsicherheit mit SSL finden Sie unter

- „IBM WebSphere MQ Unterstützung für SSL und TLS” auf Seite 25

Details zu MQSC-Befehlen, die SSL zugeordnet sind, finden Sie im Abschnitt.

- ALTER AUTHINFO
- DEFINE AUTHINFO
- DELETE AUTHINFO
- DISPLAY AUTHINFO

Ausführliche Informationen zu PCF-Befehlen im Zusammenhang mit SSL finden Sie unter

- Authentifizierungsdatenobjekt ändern, kopieren und erstellen
- Authentifizierungsdatenobjekt löschen

- Authentifizierungsdatenobjekt abfragen

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:

Windows UNIX Linux Auf Windows-, UNIX-und Linux -Systemen können Sie selbst signierte Zertifikate verwenden. Anweisungen dazu finden Sie unter „Selbst signiertes persönliches Zertifikat auf Systemen mit UNIX, Linux, and Windows erstellen” auf Seite 129.

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 kompromittiertes 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 Ihr digitales Zertifikat zu wählen. Wenn die Sicherheitsrichtlinie Ihres Unternehmens die Verwendung einer bestimmten CipherSpec erfordert, müssen Sie ein geeignetes digitales Zertifikat anfordern.

Weitere Informationen über die Beziehung zwischen CipherSpecs und digitalen Zertifikaten finden Sie in „Digitale Zertifikate und CipherSpec -Kompatibilität in IBM WebSphere MQ” auf Seite 36.

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 in WebSphere MQfinden Sie unter „Zertifikatsprüfrichtlinien in IBM WebSphere MQ” auf Seite 35.

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 einem SNA-Netz sind WebSphere MQ -MCAs 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 derselben Dokumentation finden Sie Informationen zu stärkeren Formen der Verschlüsselung, die auf Ihrer Plattform verwendet werden können, wie z. B. Triple DES-24-Byte-Verschlüsselung unter z/OS.

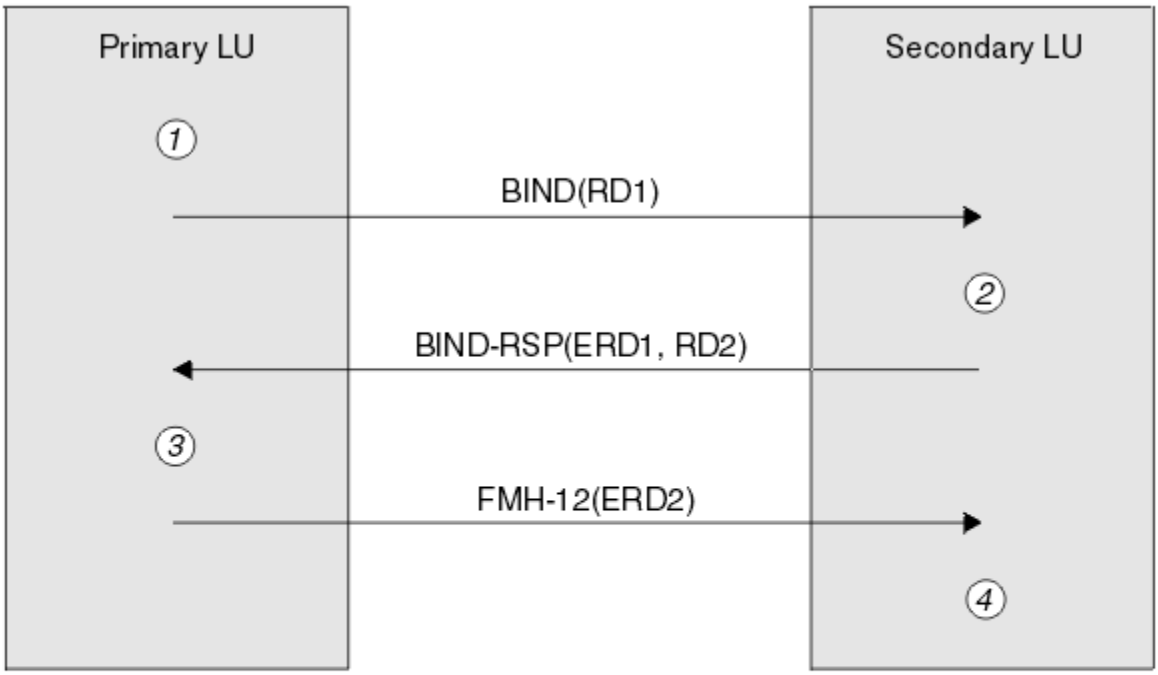
Weitere allgemeine Informationen zur Verschlüsselung auf Sitzungsebene finden Sie im Handbuch *Systems Network Architecture LU 6.2 Reference: Peer Protocols* , IBM Form SC31-6808.

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 10 auf Seite 80 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 10. 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 10 auf Seite 80.

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 WebSphere 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. Informationen zu z/OSfinden Sie in *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 in *z/OS MVS Programming: Writing Transaction Programs for APPC/MVS*, SA22-7621.

Unterstützung für die Authentifizierung auf Dialogebene in IBM WebSphere MQ auf UNIX -und Windows -Systemen
In diesem Abschnitt finden Sie eine Übersicht über die Funktionsweise der Authentifizierung auf Dialogebene unter UNIX, Linux, and Windows.

Die Unterstützung für die Authentifizierung auf Dialogebene in IBM WebSphere MQ für WebSphere MQ auf UNIX -Systemen und WebSphere MQ für Windows ist in Abbildung 11 auf Seite 82dargestellt. Die Zahlen in dem Diagramm entsprechen den Zahlen in der nachfolgenden Beschreibung.

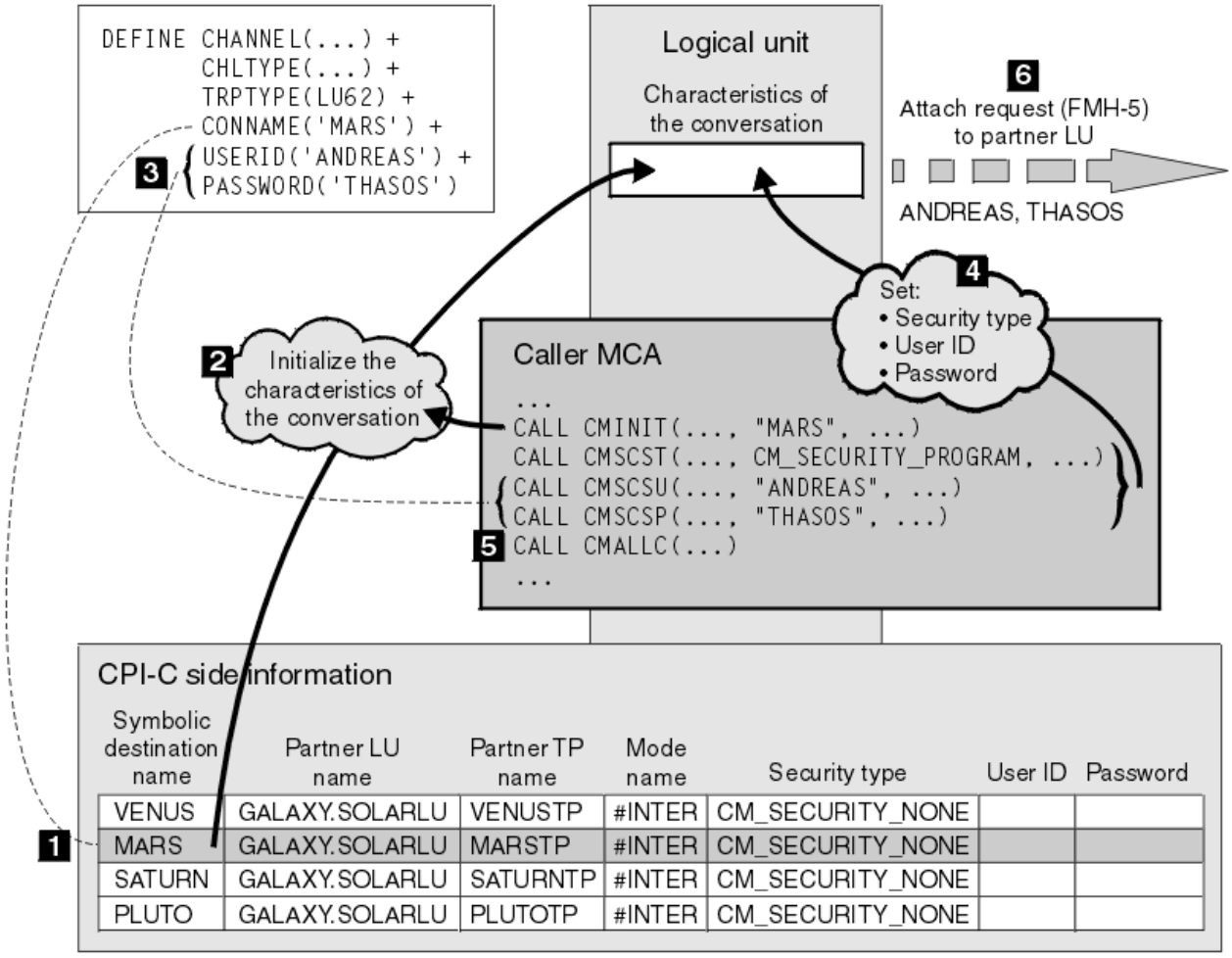


Abbildung 11. WebSphere MQ -Unterstützung für die Authentifizierung auf Dialogebene

Auf IBM i-, UNIX -und Windows -Systemen verwendet ein MCA CPI-C-Aufrufe (Common Programming Interface Communications), um über ein SNA-Netz mit einem Partner-MCA 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.

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 WebSphere MQ fest, an welchen Warteschlangenmanager sie gesendet werden soll. Wenn viele Anwendungen Nachrichten an die Clusterwarteschlange senden, verteilt WebSphere MQ die Arbeitslast auf alle Warteschlangenmanager, die über eine Instanz der Warteschlange verfügen. Wenn eines der Systeme mit einem geklonten Warteschlangenmanager ausfällt, verteilt WebSphere MQ die Arbeitslast auf die verbleibenden Warteschlangenmanager, bis das ausgefallene 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 .

Zugehörige Tasks

„Verhindern, dass Warteschlangenmanager Nachrichten empfangen” auf Seite 263
Sie können verhindern, dass ein Cluster-WS-Manager Nachrichten empfängt, die er mit Exitprogrammen nicht empfangen kann.

Sicherheit für IBM WebSphere MQ Publish/Subscribe

Es gibt zusätzliche Sicherheitsaspekte, die Sie bei der Verwendung von IBM WebSphere 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 WebSphere 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 WebSphere MQ-Publish/Subscribe bereitgestellt wird. Sie empfängt veröffentlichte Nachrichten von Subskribenten und Subskriptionsanforderun-

gen 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 WebSphere MQ Multicast unter Umständen erforderlich sind.

IBM WebSphere MQ Multicast verfügt über keine integrierte Sicherheit. Sicherheitsprüfungen werden im Warteschlangenmanager auf MQOPEN-Zeit verarbeitet, und die MQMD-Feldeinstellung wird vom Client verarbeitet. Einige Anwendungen im Netz sind möglicherweise keine IBM WebSphere MQ -Anwendungen (z. B. LLM-Anwendungen, siehe Multicast-Interoperabilität mit WebSphere MQ Low Latency Messaging für weitere Informationen). Daher müssen Sie möglicherweise Ihre eigenen Sicherheitsprozeduren implementieren, da sich die empfangenden Anwendungen nicht sicher sind, ob Kontextfelder gültig sind.

Es gibt drei Sicherheitsprozesse, die man in Betracht ziehen kann:

Zugriffssteuerung

Die Zugriffssteuerung in IBM WebSphere MQ basiert auf Benutzer-IDs. Weitere Informationen zu diesem Thema finden Sie in „Zugriffssteuerung für Clients” auf Seite 61.

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 7.

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 WebSphere

MQ vorübergehend blockieren, während Sie möglicherweise auf einen Sicherheitsadministrator warten, um die Firewallregeln zu aktualisieren.

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 193.

Sicherheit für IBM WebSphere MQ Internet Pass-thru

Internet Pass-thru kann die Kommunikation durch eine Firewall vereinfachen, aber dies hat Auswirkungen auf die Sicherheit.

IBM WebSphere MQ Internet Pass-through ist eine IBM WebSphere MQ Basisprodukterweiterung, die im SupportPac MS81bereitgestellt wird.

WebSphere MQ Internet-Pass-through ermöglicht zwei Warteschlangenmanagern den Austausch von Nachrichten oder einer WebSphere MQ -Clientanwendung, eine Verbindung zu einem Warteschlangenmanager über das Internet herzustellen, ohne dass eine direkte TCP/IP-Verbindung erforderlich ist. Dies ist nützlich, wenn eine Firewall eine direkte TCP/IP-Verbindung zwischen zwei Systemen verhindert. Dadurch wird die Übertragung des WebSphere MQ -Kanalprotokolls in eine und aus einer Firewall einfacher und einfacher zu verwalten, indem die Abläufe in HTTP oder als Proxy getunnelt werden. Bei Verwendung von Secure Sockets Layer (SSL) kann dieses Produkt auch zur Verschlüsselung bzw. Entschlüsselung der Nachrichten verwendet werden, die über das Internet gesendet werden.

Wenn Ihr WebSphere MQ -System mit IPT kommuniziert, stellen Sie sicher, dass die CipherSpec , die von WebSphere MQ verwendet wird, mit der von IPT verwendeten CipherSuite übereinstimmt, sofern Sie nicht SSLProxyMode in IPT verwenden:

- Wenn IPT als SSL-oder TLS-Server fungiert und WebSphere MQ als SSL-oder TLS-Client eine Verbindung herstellt, muss die CipherSpec , die von WebSphere MQ verwendet wird, einer CipherSuite entsprechen, die im relevanten IPT-Schlüsselring aktiviert ist.
- Wenn IPT als SSL-oder TLS-Client fungiert und eine Verbindung zu einem WebSphere MQ -SSL-oder TLS-Server herstellt, muss die IPT- CipherSuite mit der CipherSpec übereinstimmen, die im empfangenden WebSphere MQ -Channel definiert ist.

Wenn Sie eine Migration von IPT auf die integrierte SSL-und TLS-Unterstützung von WebSphere MQ durchführen, übertragen Sie die digitalen Zertifikate mithilfe von iKeyman.

Weitere Informationen finden Sie in WebSphere MQ Internet Pass-Thru (SupportPac MS81).

Sicherheit konfigurieren

Diese Themensammlung enthält spezifische Informationen zu verschiedenen Betriebssystemen und zur Verwendung von Clients.

Sicherheit auf UNIX, Linux, and Windows -Systemen einrichten

Sicherheitsaspekte, die speziell für UNIX, Linux, and Windows -Systeme gelten.

Die Warteschlangenmanager von IBM WebSphere 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 WebSphere MQ verwalten?
Sie können die Gruppe der Benutzer definieren, die Befehle für die Verwaltung von IBM WebSphere MQ ausgeben können.

Wer kann IBM WebSphere 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 auf IBM WebSphere MQ -Nachrichten zugreifen kann.
- 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 WebSphere MQ-Verzeichnis. Verwenden Sie keine Standardbefehle für das Betriebssystem, um Berechtigungen für MQI-Ressourcen zu erteilen oder zu entziehen.

Verbindung zu IBM WebSphere MQ über Terminal Services herstellen

Das **Create global objects** -Benutzerrecht kann Probleme verursachen, wenn Sie Terminal Services verwenden.

Wenn Sie über Terminal Services eine Verbindung zu einem Windows -System herstellen und Probleme beim Erstellen oder Starten eines Warteschlangenmanagers haben, kann dies auf die Benutzerberechtigung **Create global objects**in den letzten Versionen von Windowszurückzuführen sein.

Das **Create global objects** -Benutzerrecht begrenzt die Benutzer, die berechtigt sind, Objekte im globalen Namensbereich zu erstellen. Damit eine Anwendung ein globales Objekt erstellen kann, muss sie entweder im globalen Namensbereich ausgeführt werden, oder der Benutzer, unter dem die Anwendung ausgeführt wird, muss den **Create global objects** -Benutzer richtig angewendet haben.

Administratoren verfügen standardmäßig über das **Create global objects** -Benutzerrecht, so dass ein Administrator Warteschlangenmanager erstellen und starten kann, wenn er mit Terminal Services verbunden ist, ohne die Benutzerberechtigungen zu ändern.

Wenn die verschiedenen Methoden zur Verwaltung von WebSphere MQ nicht funktionieren, wenn Sie Terminalservices verwenden, versuchen Sie, die Benutzerberechtigung **Create global objects** festzulegen:

1. Öffnen Sie die Anzeige Verwaltungstools:

   Windows 2003 und Windows XP
   Rufen Sie diese Anzeige über die **Systemsteuerung** > **Verwaltung**auf.

   Windows Vista und Windows Server 2008
   Greifen Sie über die **Systemsteuerung** > **System und Wartung** > **Verwaltung**auf diese Anzeige zu.
2. Klicken Sie doppelt auf **Lokale Sicherheitsrichtlinie** .
3. Erweitern Sie `Local Policies`.
4. Klicken Sie auf `User Rights Assignment`.
5. Fügen Sie den neuen Benutzer oder die neue Gruppe zur Richtlinie **Create global objects** hinzu.

Gruppen unter Windows erstellen und verwalten

Diese Anweisungen führen Sie durch den Prozess der Verwaltung von Gruppen auf einer Workstation oder einer Mitgliedsservermaschine.

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 Principals vornehmen, werden erst erkannt, wenn der Warteschlangenmanager erneut gestartet wird, oder Sie setzen den MQSC-Befehl REFRESH SECURITY (oder die PCF-Entsprechung) ab.

Verwenden Sie die Anzeige 'Computerverwaltung', um mit Benutzer und Gruppen zu arbeiten. Alle Änderungen am aktuellen angemeldeten Benutzer sind möglicherweise erst dann wirksam, wenn sich der Benutzer erneut anmeldet.

Windows 2003 und Windows XP
Greifen Sie über **Systemsteuerung** > **Verwaltung** > **Computerverwaltung**auf diese Anzeige zu.

Windows Vista und Windows Server 2008
Greifen Sie über die **Systemsteuerung** > **System und Wartung** > **Verwaltung** > **Computerverwaltung**auf diese Anzeige zu.

Windows 7
Über **Verwaltungstools** > **Computerverwaltung** auf diese Anzeige zugreifen

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** .

Unter Windows einer Gruppe einen Benutzer 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.

Anzeigen, wer sich in einer Gruppe unter Windows befindet

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.

Entfernen eines Benutzers aus einer Gruppe unter Windows

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.

Gruppen unter HP-UX erstellen und verwalten

Wenn Sie unter HP-UXnicht NIS oder NIS + verwenden, verwenden Sie System Administration Manager (SAM), um mit Gruppen zu arbeiten.

Gruppe unter HP-UX erstellen
Benutzer zu einer Gruppe hinzufügen, indem der Systemverwaltungsmanager verwendet wird

Vorgehensweise

1. Klicken Sie im Systemverwaltungsmanager doppelt auf 'Accounts for Users and Groups' (Konten für Benutzer und Gruppen).
2. Klicken Sie doppelt auf 'Groups' (Gruppen).
3. Wählen Sie Add from the Actions pull down to display the Add a New Group panel aus.
4. Geben Sie den Namen der Gruppe ein und wählen Sie die Benutzer aus, die der Gruppe hinzugefügt werden sollen.
5. Klicken Sie auf Anwenden, um die Gruppe zu erstellen.

Ergebnisse
Sie haben nun eine Gruppe erstellt.

Benutzer unter HP-UX zu einer Gruppe hinzufügen
Fügen Sie einen Benutzer zu einer Gruppe hinzu, indem Sie den Systemverwaltungsmanager verwenden.

Vorgehensweise

1. Klicken Sie im Systemverwaltungsmanager doppelt auf 'Accounts for Users and Groups' (Konten für Benutzer und Gruppen).
2. Klicken Sie doppelt auf 'Groups' (Gruppen).
3. Heben Sie den Namen der Gruppe hervor, und wählen Sie im Pulldown-Menü "Aktionen" die Option Ändern aus, um die Anzeige "Vorhandene Gruppe ändern" anzuzeigen.
4. Wählen Sie einen Benutzer aus, der der Gruppe hinzugefügt werden soll, und klicken Sie auf Hinzufügen.
5. Wenn Sie weitere Benutzer zur Gruppe hinzufügen möchten, wiederholen Sie Schritt 4 für jeden Benutzer.
6. Wenn Sie alle Namen zur Liste hinzugefügt haben, klicken Sie auf OK.

Ergebnisse
Sie haben nun einen Benutzer zu einer Gruppe hinzugefügt.

Mitglieder in einer Gruppe unter HP-UX anzeigen
Anzeigen, wer sich in einer Gruppe befindet, indem er den Systemverwaltungsmanager verwendet

Vorgehensweise

1. Klicken Sie im Systemverwaltungsmanager doppelt auf 'Accounts for Users and Groups' (Konten für Benutzer und Gruppen).
2. Klicken Sie doppelt auf 'Groups' (Gruppen).
3. Heben Sie den Namen der Gruppe hervor, und wählen Sie im Pulldown-Menü "Aktionen" die Option Ändern aus, um die Anzeige "Vorhandene Gruppe ändern" aufzurufen, in der eine Liste der Benutzer in der Gruppe angezeigt wird.

Ergebnisse
Die Gruppenmitglieder werden angezeigt.

Benutzer unter HP-UX aus einer Gruppe entfernen
Entfernen Sie einen Benutzer aus einer Gruppe, indem Sie den Systemverwaltungsmanager verwenden.

Vorgehensweise

1. Klicken Sie im Systemverwaltungsmanager doppelt auf 'Accounts for Users and Groups' (Konten für Benutzer und Gruppen).
2. Klicken Sie doppelt auf 'Groups' (Gruppen).
3. Heben Sie den Namen der Gruppe hervor, und wählen Sie im Pulldown-Menü "Aktionen" die Option Ändern aus, um die Anzeige "Vorhandene Gruppe ändern" anzuzeigen.
4. Wählen Sie einen Benutzer aus, der aus der Gruppe entfernt werden soll, und klicken Sie auf Entfernen.
5. Wenn Sie andere Benutzer aus der Gruppe entfernen möchten, wiederholen Sie Schritt 4 für jeden Benutzer.
6. Wenn Sie die Namen aus der Liste entfernt haben, klicken Sie auf OK.

Ergebnisse
Sie haben nun einen Benutzer aus einer Gruppe entfernt.

Gruppen unter AIX erstellen und verwalten

Wenn Sie unter AIXnicht NIS oder NIS + verwenden, verwenden Sie SMITTY, um mit Gruppen zu arbeiten.

Erstellen einer Gruppe
Erstellen Sie eine Gruppe mit SMITTY.

Vorgehensweise

1. Wählen Sie in SMITTY Security and Users aus, und drücken Sie die Eingabetaste.
2. Wählen Sie Gruppen aus, und drücken Sie die Eingabetaste
3. Wählen Sie Add a Group aus, und drücken Sie die Eingabetaste.
4. Geben Sie den Namen der Gruppe und die Namen der Benutzer ein, die der Gruppe hinzugefügt werden sollen, getrennt durch Kommas.
5. Drücken Sie die Eingabetaste, um die Gruppe zu erstellen.

Ergebnisse
Sie haben nun eine Gruppe erstellt.

Benutzer zu einer Gruppe hinzufügen
Fügen Sie einen Benutzer zu einer Gruppe hinzu, indem Sie SMITTY verwenden.

Vorgehensweise

1. Wählen Sie in SMITTY Security and Users aus, und drücken Sie die Eingabetaste.
2. Wählen Sie Gruppen aus, und drücken Sie die Eingabetaste
3. Wählen Sie Change/Show Characteristics of Groups aus, und drücken Sie die Eingabetaste.
4. Geben Sie den Namen der Gruppe ein, um eine Liste der Mitglieder der Gruppe anzuzeigen.
5. Fügen Sie die Namen der Benutzer, die der Gruppe hinzugefügt werden sollen, durch Kommas getrennt hinzu.
6. Drücken Sie die Eingabetaste, um die Namen der Gruppe hinzuzufügen.

Mitglieder einer Gruppe anzeigen

Anzeigen, wer sich in einer Gruppe mit SMITTY befindet.

Vorgehensweise

1. Wählen Sie in SMITTY Security and Users aus, und drücken Sie die Eingabetaste.
2. Wählen Sie Gruppen aus, und drücken Sie die Eingabetaste
3. Wählen Sie Change/Show Characteristics of Groups aus, und drücken Sie die Eingabetaste.
4. Geben Sie den Namen der Gruppe ein, um eine Liste der Mitglieder der Gruppe anzuzeigen.

Ergebnisse

Die Gruppenmitglieder werden angezeigt.

Benutzer aus einer Gruppe entfernen

Entfernen Sie einen Benutzer aus einer Gruppe, indem Sie SMITTY verwenden.

Vorgehensweise

1. Wählen Sie in SMITTY Security and Users aus, und drücken Sie die Eingabetaste.
2. Wählen Sie Gruppen aus, und drücken Sie die Eingabetaste
3. Wählen Sie Change/Show Characteristics of Groups aus, und drücken Sie die Eingabetaste.
4. Geben Sie den Namen der Gruppe ein, um eine Liste der Mitglieder der Gruppe anzuzeigen.
5. Löschen Sie die Namen der Benutzer, die aus der Gruppe entfernt werden sollen.
6. Drücken Sie die Eingabetaste, um die Namen aus der Gruppe zu entfernen.

Ergebnisse

Sie haben nun einen Benutzer aus einer Gruppe entfernt.

Gruppen unter Solaris erstellen und verwalten

Wenn Sie unter Solaris nicht NIS oder NIS + verwenden, verwenden Sie die Datei `/etc/group` , um mit Gruppen zu arbeiten.

Gruppe unter Solaris erstellen

Gruppe mit dem Befehl **groupadd** erstellen.

Vorgehensweise

Geben Sie den folgenden Befehl ein: `groupadd` *`group-name`*

Dabei steht *Gruppenname* für den Namen der Gruppe.

Ergebnisse

Die Datei `/etc/group` enthält Gruppeninformationen.

Benutzer unter Solaris zu einer Gruppe hinzufügen

Verwenden Sie den Befehl **usermod** , um einen Benutzer zu einer Gruppe hinzuzufügen.

Vorgehensweise

Wenn Sie eine Teildatei zu einer zusätzlichen Gruppe hinzufügen möchten, führen Sie den Befehl `usermod` aus und listen Sie die ergänzenden Gruppen auf, zu denen der Benutzer derzeit gehört, und die ergänzenden Gruppen, zu denen der Benutzer gehören soll.

Wenn der Benutzer beispielsweise Mitglied der Gruppe groupaist und auch Mitglied von groupb werden soll, verwenden Sie den Befehl `usermod -G groupa,groupb user-name` , wobei *Benutzername* der Benutzername ist.

Anzeigen, wer sich in einer Gruppe unter Solaris befindet

Wenn Sie feststellen möchten, wer Mitglied einer Gruppe ist, sehen Sie sich den Eintrag für diese Gruppe in der Datei `/etc/group` an.

Benutzer unter Solaris aus einer Gruppe entfernen

Verwenden Sie den Befehl **usermod** , um einen Benutzer aus einer Gruppe zu entfernen.

Vorgehensweise

Um ein Mitglied aus einer ergänzenden Gruppe zu entfernen, führen Sie den Befehl **usermod** aus, der die ergänzenden Gruppen auflistet, deren Mitglied der Benutzer bleiben soll.
Wenn die Primärgruppe des Benutzers z. B. `users` ist und der Benutzer auch Mitglied der Gruppen `mqm`, `groupa` und `groupb` ist, um den Benutzer aus der Gruppe `mqm` zu entfernen, wird der folgende Befehl verwendet: `usermod -G groupa,groupb user-name`, wobei *user-name* der Benutzername ist.

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`.

Gruppe unter Linux erstellen

Erstellen Sie eine Gruppe mit dem Befehl **groupadd** .

Vorgehensweise

Geben Sie den folgenden Befehl ein, um eine neue Gruppe zu erstellen: `groupadd -g group-ID group-name`

, wobei *group-ID* für die numerische Kennung der Gruppe steht und *group-name* der Name der Gruppe ist.

Ergebnisse

Die Datei `/etc/group` enthält Gruppeninformationen.

Benutzer unter Linux einer Gruppe hinzufügen

Verwenden Sie den Befehl **usermod** , um einen Benutzer zu einer Gruppe hinzuzufügen.

Vorgehensweise

Wenn Sie eine Teildatei zu einer zusätzlichen Gruppe hinzufügen möchten, führen Sie den Befehl `usermod` aus und listen Sie die ergänzenden Gruppen auf, zu denen der Benutzer derzeit gehört, und die ergänzenden Gruppen, zu denen der Benutzer gehören soll.
Wenn der Benutzer z. B. Mitglied der Gruppe `groupa` ist und auch Mitglied von `groupb` werden soll, wird der folgende Befehl verwendet: `usermod -G groupa,groupb user-name`

, wobei *user-name* für den Benutzernamen steht.

Mitglieder in einer Gruppe unter Linux anzeigen

Zeigen Sie mit dem Befehl **getent** an, wer sich in einer Gruppe befindet.

Vorgehensweise

Geben Sie den folgenden Befehl ein, um anzuzeigen, wer Mitglied einer Gruppe ist: `getent group group-name`

, wobei *group-name* für den Namen der Gruppe steht.

Benutzer aus einer Gruppe entfernen

Verwenden Sie den Befehl **usermod** , um einen Benutzer aus einer Gruppe zu entfernen.

Vorgehensweise

Um ein Mitglied aus einer ergänzenden Gruppe zu entfernen, führen Sie den Befehl **usermod** aus, der die ergänzenden Gruppen auflistet, deren Mitglied der Benutzer bleiben soll.
Wenn beispielsweise die Primärgruppe des Benutzers `users` ist und der Benutzer auch Mitglied der Gruppen `mqm`, `groupa` und `groupbist`, wird der Benutzer aus der Gruppe `mqm` entfernt, der folgende Befehl verwendet: `usermod -G groupa,groupb` *`user-name`*

, wobei *user-name* für den Benutzernamen steht.

Funktionsweise der Autorisierungen

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.

Die in den Tabellen angegebenen Konstanten mit dem Präfix MQZAO_ entsprechen den Schlüsselwörtern in der Berechtigungsliste des Befehls `setmqaut` für die jeweilige 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.

Berechtigungen für MQI-Aufrufe

MQCONN, **MQOPEN**, **MQPUT1**und **MQCLOSE** erfordern möglicherweise Berechtigungsprüfungen. 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.

Vier MQI-Aufrufe erfordern möglicherweise Berechtigungsprüfungen: **MQCONN**, **MQOPEN**, **MQPUT1**und **MQCLOSE**.

Für MQOPEN und MQPUT1erfolgt die Berechtigungsprüfung für den Namen des zu öffnenden Objekts und nicht für den Namen oder die Namen, die sich nach der Auflösung 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 für die erste Definition durchgeführt wird, die während des Prozesses zum Auflösen eines Namens gefunden wird, der kein Warteschlangenmanager-Aliasname ist, es sei denn, die Warteschlangenmanager-Aliasdefinition wird direkt geöffnet, d. h., ihr 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.

Tabelle 8 auf Seite 95, Tabelle 9 auf Seite 95, Tabelle 10 auf Seite 96 und Tabelle 11 auf Seite 96 fassen die Berechtigungen zusammen, die für die einzelnen Telefonanrufe benötigt werden. 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 8. Für MQCONN-Aufrufe erforderliche Sicherheitsberechtigung

Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 96)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQCONN	Nicht zutreffend	Nicht zutreffend	MQZAO_CONNECT

Tabelle 9. Für MQOPEN-Aufrufe erforderliche Sicherheitsberechtigung

Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 96)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQOO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE
MQOO_BROWSE	MQZAO_BROWSE	Nicht zutreffend	Keine Prüfung
MQOO_INPUT_*	MQZAO_INPUT	Nicht zutreffend	Keine Prüfung
MQOO_SAVE_ ALL_CONTEXT **(„2” auf Seite 97)**	MQZAO_INPUT	Nicht zutreffend	Nicht zutreffend
MQOO_OUTPUT (normale Warteschlange) **(„3” auf Seite 97)**	MQZAO_OUTPUT	Nicht zutreffend	Nicht zutreffend
MQOO_PASS_ IDENTITY_CONTEXT **(„4” auf Seite 97)**	MQZAO_PASS_ IDENTITY_CONTEXT	Nicht zutreffend	Keine Prüfung
MQOO_PASS_ALL_ CONTEXT **(„4” auf Seite 97, „5” auf Seite 97)**	MQZAO_PASS _ALL_CONTEXT	Nicht zutreffend	Keine Prüfung
MQOO_SET_ IDENTITY_CONTEXT **(„4” auf Seite 97, „5” auf Seite 97)**	MQZAO_SET_ IDENTITY_CONTEXT	Nicht zutreffend	MQZAO_SET_ IDENTITY_CONTEXT **(„6” auf Seite 97)**
MQOO_SET_ ALL_CONTEXT **(„4” auf Seite 97, „7” auf Seite 97)**	MQZAO_SET_ ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ ALL_CONTEXT **(„6” auf Seite 97)**

Tabelle 9. Für MQOPEN-Aufrufe erforderliche Sicherheitsberechtigung (Forts.)			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 96)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQOO_OUTPUT (Übertragungswarteschlange) **(„8” auf Seite 97)**	MQZAO_SET_ ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ ALL_CONTEXT **(„6” auf Seite 97)**
MQOO_SET	MQZAO_SET	Nicht zutreffend	Keine Prüfung
MQOO_ALTERNATE_ USER_AUTHORITY	**(„9” auf Seite 97)**	**(„9” auf Seite 97)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(„9” auf Seite 97, „10” auf Seite 97)**

Tabelle 10. Für MQPUT1-Aufrufe erforderliche Sicherheitsberechtigung			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 96)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(„11” auf Seite 97)**	Nicht zutreffend	Keine Prüfung
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(„11” auf Seite 97)**	Nicht zutreffend	Keine Prüfung
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(„11” auf Seite 97)**	Nicht zutreffend	MQZAO_SET_ IDENTITY_CONTEXT **(„6” auf Seite 97)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(„11” auf Seite 97)**	Nicht zutreffend	MQZAO_SET_ ALL_CONTEXT **(„6” auf Seite 97)**
(Übertragungswarteschlange) **(„8” auf Seite 97)**	MQZAO_SET_ ALL_CONTEXT	Nicht zutreffend	MQZAO_SET_ ALL_CONTEXT **(„6” auf Seite 97)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(„12” auf Seite 97)**	Nicht zutreffend	MQZAO_ALTERNATE_ USER_AUTHORITY **(„10” auf Seite 97)**

Tabelle 11. Für MQCLOSE-Aufrufe erforderliche Sicherheitsberechtigung			
Erforderliche Berechtigung für:	**Warteschlangenobjekt („1” auf Seite 96)**	**Prozessobjekt**	**WS-Manager-Objekt**
MQCO_DELETE	MQZAO_DELETE **(„13” auf Seite 97)**	Nicht zutreffend	Nicht zutreffend
MQCO_DELETE _PURGE	MQZAO_DELETE **(„13” auf Seite 97)**	Nicht zutreffend	Nicht zutreffend

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 97).
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 ausgeführt, wenn die beiden folgenden Bedingungen zutreffen:
    - Eine permanente dynamische Warteschlange wird geschlossen und gelöscht.
    - Die Warteschlange wurde nicht vom Aufruf MQOPEN erstellt, der die verwendete Objektkennung zurückgegeben hat.

    Sonst gibt es keine Prüfung.

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 *Objekt*

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 *Objekt*

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
Kommunikationsinformationen	Nicht zutreffend

DEFINE *Objekt* NOREPLACE („1" auf Seite 102)

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(„2" auf Seite 102)**
Thema	MQZAO_CREATE **(„2" auf Seite 102)**
Prozess	MQZAO_CREATE **(„2" auf Seite 102)**
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_CREATE **(„2" auf Seite 102)**
Authentifizierungsdaten	MQZAO_CREATE **(„2" auf Seite 102)**
Kanal	MQZAO_CREATE **(„2" auf Seite 102)**

Objekt	Erforderliche Berechtigung
Clientverbindungskanal	MQZAO_CREATE **(„2” auf Seite 102)**
Empfangsprogramm	MQZAO_CREATE **(„2” auf Seite 102)**
Service	MQZAO_CREATE **(„2” auf Seite 102)**
Kommunikationsinformationen	MQZAO_CREATE **(„2” auf Seite 102)**

DEFINE *object* REPLACE („1” auf Seite 102, „3” auf Seite 102)

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
Kommunikationsinformationen	MQZAO_ÄNDERUNG

DELETE *Objekt*

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 *Objekt*

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DISPLAY
Thema	MQZAO_DISPLAY

Objekt	Erforderliche Berechtigung
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
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

Objekt	Erforderliche Berechtigung
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
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

Befehl	Objekt	Erforderliche Berechtigung
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 eines angegebenen Warteschlangenmanagers erteilt, indem der 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

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.

***Objekt* ändern**

Objekt	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

***Objekt* leeren**

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
Kommunikationsinformationen	Nicht zutreffend

***Objekt* kopieren (ohne Ersetzen) (1)**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(2)**
Thema	MQZAO_CREATE **(2)**
Prozess	MQZAO_CREATE **(2)**
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_CREATE **(2)**
Authentifizierungsinformationen	MQZAO_CREATE **(2)**
Kanal	MQZAO_CREATE **(2)**

Objekt	Erforderliche Berechtigung
Clientverbindungskanal	MQZAO_CREATE **(2)**
Listener	MQZAO_CREATE **(2)**
Service	MQZAO_CREATE **(2)**
Kommunikationsinformationen	MQZAO_CREATE **(„2” auf Seite 109)**

Kopieren *Objekt* (mit Ersetzen) (1, 4)

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsinformationen	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Listener	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG
Kommunikationsinformationen	MQZAO_ÄNDERUNG

***Objekt* (ohne Ersetzen) (3) erzeugen.**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_CREATE **(2)**
Thema	MQZAO_CREATE **(2)**
Prozess	MQZAO_CREATE **(2)**
Warteschlangenmanager	Nicht zutreffend
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)**

***Objekt* (mit Ersetzen) erstellen (3, 4)**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_ÄNDERUNG
Thema	MQZAO_ÄNDERUNG

Objekt	Erforderliche Berechtigung
Prozess	MQZAO_ÄNDERUNG
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_ÄNDERUNG
Authentifizierungsinformationen	MQZAO_ÄNDERUNG
Kanal	MQZAO_ÄNDERUNG
Clientverbindungskanal	MQZAO_ÄNDERUNG
Listener	MQZAO_ÄNDERUNG
Service	MQZAO_ÄNDERUNG
Kommunikationsinformationen	MQZAO_ÄNDERUNG

***Objekt* löschen**

Objekt	Erforderliche Berechtigung
Warteschlange	MQZAO_DELETE
Thema	MQZAO_DELETE
Prozess	MQZAO_DELETE
Warteschlangenmanager	Nicht zutreffend
Namensliste	MQZAO_DELETE
Authentifizierungsinformationen	MQZAO_DELETE
Kanal	MQZAO_DELETE
Clientverbindungskanal	MQZAO_DELETE
Listener	MQZAO_DELETE
Service	MQZAO_DELETE
Kommunikationsinformationen	MQZAO_DELETE

***Objekt* abfragen**

Objekt	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

Objekt	Erforderliche Berechtigung
Kommunikationsinformationen	MQZAO_DISPLAY

***Objekt*namen abfragen**

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
Kommunikationsinformationen	Keine Prüfung

***object* starten**

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
Listener	MQZAO_CONTROL
Service	MQZAO_CONTROL
Kommunikationsinformationen	Nicht zutreffend

***object* stoppen**

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
Listener	MQZAO_CONTROL
Service	MQZAO_CONTROL
Kommunikationsinformationen	Nicht zutreffend

Kanalbefehle

Befehl	Objekt	Erforderliche Berechtigung
Pingkanal	Kanal	MQZAO_CONTROL
Kanal zurücksetzen	Kanal	MQZAO_CONTROL_EXTENDED
Auflösungskanal	Kanal	MQZAO_CONTROL_EXTENDED

Subskriptionsbefehle

Befehl	Objekt	Erforderliche Berechtigung
Subskription ändern	Thema	MQZAO_CONTROL
Subskription erstellen	Thema	MQZAO_CONTROL
Subskription löschen	Thema	MQZAO_CONTROL
Inquire Subscription	Thema	MQZAO_DISPLAY

Sicherheitsbefehle

Befehl	Objekt	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	Objekt	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	Objekt	Erforderliche Berechtigung
Clusterwarteschlangenmanager anfragen	Warteschlangenmanager	MQZAO_DISPLAY
Cluster aktualisieren	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	
Cluster zurücksetzen	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	
Clusterwarteschlangenmanager-Cluster aussetzen	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	
WS-Manager-Cluster wieder aufnehmen	Erforderliche Gruppenzugehörigkeit 'mqm' erforderlich	

Andere Verwaltungsbefehle

Befehl	Objekt	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 eines angegebenen Warteschlangenmanagers erteilt, indem der Objekttyp QMGR im Befehl `setmqaut` angegeben wird.
3. Für Erstellungsbefehle ist auch die Berechtigung MQZAO_DISPLAY für das entsprechende SYSTEM.DEFAULT.* Objekt.
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.

Besondere Hinweise zur Sicherheit unter Windows

Einige Sicherheitsfunktionen verhalten sich bei verschiedenen Versionen von Windowsunterschiedlich.

Die IBM WebSphere MQ-Sicherheit basiert auf Aufrufen an die Betriebssystem-API, in denen Informationen zu Benutzerberechtigungen und Gruppenzugehörigkeit angefordert werden. Einige Funktionen verhalten sich auf Windows -Systemen nicht identisch. Diese Themensammlung enthält Beschreibungen, wie sich diese Unterschiede auf die IBM WebSphere MQ -Sicherheit auswirken können, wenn Sie IBM WebSphere MQ in einer Windows -Umgebung ausführen.

Das SSPI-Kanalexitprogramm

WebSphere MQ for Windows stellt ein Sicherheitsexitprogramm bereit, das sowohl für Nachrichten-als auch für MQI-Kanäle 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 SSPI (Security Support Provider Interface), die die integrierten Sicherheitsfunktionen von Windows -Plattformen bereitstellt.

Der Sicherheitsexit stellt die folgenden Identifizierungs-und Authentifizierungsservices bereit:

Einweg-Authentifizierung
Dies verwendet die Unterstützung der Windows NT LAN Manager-Authentifizierung (NTLM-Authentifizierung). 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 WebSphere MQ Version 7.0unterstützt werden.

Dieser Service wird normalerweise in einem MQI-Kanal verwendet, um einem Server-Warteschlangenmanager die Authentifizierung einer WebSphere MQ MQI-Clientanwendung 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 WebSphere MQ Version 7.0unterstützt 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 ermöglicht es dem Server-Warteschlangenmanager und der WebSphere MQ MQI-Clientanwendung, sich gegenseitig zu 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 Sicherheits-

nachricht 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 sowie Anweisungen zur Implementierung dieses Programms finden Sie im Abschnitt SSPI-Sicherheitsexit auf Windows-Systemen verwenden.

Wenn Sie unter Windows den Fehler 'Gruppe nicht gefunden' erhalten

Dieses Problem kann auftreten, weil WebSphere MQ den Zugriff auf die lokale Gruppe 'mqm' verliert, wenn Windows -Server auf Domänencontroller hochgestuft oder von diesen herabgestuft werden. Um dieses Problem zu beheben, erstellen Sie die lokale mqm-Gruppe erneut.

Das Symptom ist ein Fehler, der den Mangel an einer lokalen mqm-Gruppe angibt, z. B.:

```
>crtmqm qm0
AMQ8066:Local mqm group not found.
```

Das Ändern des Status einer Maschine zwischen Server und Domänencontroller kann sich auf den Betrieb von WebSphere MQauswirken, da WebSphere MQ eine lokal definierte mqm-Gruppe verwendet. 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.

Zur Behebung dieses Problems erstellen Sie die lokale Gruppe 'mqm' mit den Standardverwaltungstools von Windows erneut. Da alle Informationen zur Gruppenzugehörigkeit verloren gehen, müssen Sie privilegierte WebSphere MQ -Benutzer in der neu erstellten lokalen Gruppe 'mqm' wiederherstellen. Wenn die Maschine ein Domänenmitglied ist, müssen Sie auch die Gruppe 'mqm' der lokalen Gruppe 'mqm' hinzufügen, um der privilegierten Domäne WebSphere MQ -Benutzer-IDs die erforderliche Berechtigungsstufe zu erteilen.

Wenn Probleme mit IBM WebSphere MQ und Domänencontrollern unter Windows auftreten

Bestimmte Probleme können bei Sicherheitseinstellungen auftreten, wenn Windows -Server auf Domänencontroller hochgestuft werden.

Beim Hochstufen von Windows 2000, Windows 2003 oder Windows Server 2008-Servern auf Domänencontroller wird Ihnen die Option angezeigt, eine Standardsicherheitseinstellung oder eine andere Sicherheitseinstellung für Benutzer-und Gruppenberechtigungen auszuwählen. Mit dieser Option wird gesteuert, ob beliebige Benutzer Gruppenzugehörigkeiten aus dem aktiven Verzeichnis abrufen können. Da WebSphere MQ bei der Implementierung seiner Sicherheitsrichtlinie auf Informationen zur Gruppenzugehörigkeit angewiesen ist, ist es wichtig, dass die Benutzer-ID, die WebSphere MQ -Operationen ausführt, die Gruppenzugehörigkeit anderer Benutzer bestimmen kann.

Wenn unter Windows 2000 eine Domäne mit der Standardsicherheitsoption erstellt wird, kann die von WebSphere MQ während des Installationsprozesses erstellte Standardbenutzer-ID bei Bedarf Gruppenzugehörigkeiten für andere Benutzer abrufen. Das Produkt wird dann normal installiert, wobei Standardobjekte erstellt werden, und der Warteschlangenmanager kann die Zugriffsberechtigung für lokale Benutzer und Domänenbenutzer bei Bedarf festlegen.

Wenn unter Windows 2000 eine Domäne mit der vom Standard abweichenden Sicherheitsoption erstellt wird oder unter Windows 2003 und Windows Server 2008 mit der Standardsicherheitsoption erstellt wird, kann die Benutzer-ID, die von WebSphere MQ während der Installation erstellt wird, nicht immer die erforderlichen Gruppenzugehörigkeiten bestimmen. In diesem Fall müssen Sie Folgendes wissen:

- Verhalten von Windows 2000 mit vom Standard abweichenden Berechtigungen oder Windows 2003 und Windows Server 2008 mit Standardberechtigungen
- Wie kann die Gruppe mqm-Gruppenmitglieder die Gruppenzugehörigkeit lesen?
- IBM WebSphere MQ Windows -Service für die Ausführung unter einem Domänenbenutzer konfigurieren

Windows 2000-Domäne mit vom Standard abweichenden Berechtigungen oder Windows 2003-und Windows Server 2008-Domäne mit Standardsicherheitsberechtigungen
Je nachdem, ob die Installation von einem lokalen Benutzer oder einem Domänenbenutzer ausgeführt wird, verhält sich die Installation von WebSphere MQ auf diesen Betriebssystemen unterschiedlich.

Wenn ein **lokaler** Benutzer WebSphere MQinstalliert, stellt der WebSphere MQ -Vorbereitungsassistent fest, dass der für den IBM WebSphere MQ Windows -Service erstellte lokale Benutzer die Informationen zur Gruppenzugehörigkeit des installierenden Benutzers abrufen kann. Der Assistent zur Vorbereitung von WebSphere MQ stellt dem Benutzer Fragen zur Netzkonfiguration, um festzustellen, ob auf Domänencontrollern unter Windows 2000 oder höher andere Benutzerkonten definiert sind. Wenn dies der Fall ist, muss der IBM WebSphere MQ Windows -Service unter einem Domänenbenutzerkonto mit bestimmten Einstellungen und Berechtigungen ausgeführt werden. Der Assistent zur Vorbereitung von WebSphere MQ fordert den Benutzer zur Eingabe der Kontodetails dieses Benutzers auf. Die Onlinehilfe enthält Details zu dem Domänenbenutzerkonto, das an den Domänenadministrator gesendet werden kann.

Wenn ein Benutzer der **Domäne** WebSphere MQinstalliert, erkennt der Assistent zur Vorbereitung von WebSphere MQ , dass der lokale Benutzer, der für den IBM WebSphere MQ Windows -Service erstellt wurde, die Informationen zur Gruppenzugehörigkeit des installierenden Benutzers nicht abrufen kann. In diesem Fall fordert der Assistent zur Vorbereitung von WebSphere MQ den Benutzer immer zur Eingabe der Kontodetails des Domänenbenutzeraccounts für den zu verwendenden IBM WebSphere MQ Windows -Service auf.

Wenn der IBM WebSphere MQ Windows -Dienst ein Domänenbenutzerkonto verwenden muss, kann WebSphere MQ erst ordnungsgemäß funktionieren, wenn dies mit dem WebSphere MQ -Vorbereitungsassistenten konfiguriert wurde. Der Assistent zur Vorbereitung von WebSphere MQ lässt die Fortsetzung anderer Tasks durch den Benutzer erst zu, wenn der Windows -Service mit einem geeigneten Konto konfiguriert wurde.

Wenn eine Windows 2000-Domäne mit nicht standardmäßigen Sicherheitsberechtigungen konfiguriert wurde, besteht die übliche Lösung für die ordnungsgemäße Funktion von WebSphere MQ darin, die Domäne wie oben beschrieben mit einem geeigneten Domänenbenutzerkonto zu konfigurieren.

Weitere Informationen finden Sie unter Domänenkonten für WebSphere MQ .

IBM WebSphere MQ -Services für die Ausführung unter einem Domänenbenutzer unter Windows konfigurieren
Verwenden Sie den IBM WebSphere MQ -Vorbereitungsassistenten, um die Kontodetails des Domänenbenutzerkontos einzugeben. Alternativ können Sie die Anzeige 'Computerverwaltung' verwenden, um die **Anmeldedetails** für den installationsspezifischen IBM WebSphere MQ -Service zu ändern.

Weitere Informationen finden Sie unter Kennwort des IBM WebSphere MQ Windows -Servicebenutzeraccounts ändern .

Sicherheitsvorlagendateien auf Windows anwenden

Das Anwenden einer Vorlage kann sich auf die Sicherheitseinstellungen auswirken, die auf WebSphere MQ -Dateien und -Verzeichnisse angewendet wurden. Wenn Sie die hochsichere Vorlage verwenden, wenden Sie sie an, bevor Sie WebSphere MQinstallieren.

Windows unterstützt textbasierte Sicherheitsvorlagendateien, mit deren Hilfe Sie einheitliche Sicherheitseinstellungen auf einen oder mehrere Computer mit dem MMC-Snap-in für Sicherheitskonfiguration und -analyse anwenden können. Insbesondere stellt Windows mehrere Vorlagen bereit, die eine Reihe von Si-

cherheitseinstellungen enthalten, um bestimmte Sicherheitsstufen bereitzustellen. Zu diesen Schablonen gehören Compatible, Secure und Highly Secure.

Das Anwenden einer dieser Schablonen kann sich auf die Sicherheitseinstellungen auswirken, die auf WebSphere MQ -Dateien und -Verzeichnisse angewendet wurden. Wenn Sie die Vorlage "Highly Secure" verwenden möchten, konfigurieren Sie Ihre Maschine, bevor Sie WebSphere MQinstallieren.

Wenn Sie die hochsichere Vorlage auf eine Maschine anwenden, auf der WebSphere MQ bereits installiert ist, werden alle Berechtigungen, die Sie für die WebSphere MQ -Dateien und -Verzeichnisse festgelegt haben, entfernt. Da diese Berechtigungen entfernt werden, verlieren Sie *Administrator* , *mqm* und, falls zutreffend, den Gruppenzugriff *Jeder* aus den Fehlerverzeichnissen.

Verschachtelte Gruppen

Es gibt Einschränkungen bei der Verwendung von verschachtelten Gruppen. Diese ergeben sich teilweise aus der Domänenfunktionsebene und teilweise aus WebSphere MQ -Einschränkungen.

Active Directory kann verschiedene Gruppentypen in einem Domänenkontext unterstützen, abhängig von der Domänenfunktionsebene. Standardmäßig befinden sich Windows 2003-Domänen auf der Funktionsebene *Windows 2000 gemischt* . (Windows Server 2003, Windows XP, Windows Vista und Windows Server 2008 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 Active Directory -Anforderungen gelten weitere Einschränkungen für IDs, die von WebSphere MQverwendet werden. Die von WebSphere MQ verwendeten Netz-APIs unterstützen nicht alle Konfigurationen, die von der Domänenfunktionsebene unterstützt werden. Aus diesem Grund ist WebSphere MQ nicht in der Lage, die Gruppenzugehörigkeiten von Domänen-IDs, die in einer lokalen Domänengruppe vorhanden sind, abzufragen, die dann 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.

Zusätzliche Berechtigung für Windows -Anwendungen konfigurieren, die eine Verbindung zu IBM WebSphere MQ herstellen

Für das Konto, unter dem IBM WebSphere MQ-Prozesse ausgeführt werden, sind eventuell zusätzliche Berechtigungen erforderlich, damit SYNCHRONIZE-Zugriff auf Anwendungsprozesse erteilt werden kann.

Es können Probleme auftreten, wenn Sie über Windows -Anwendungen (z. B. ASP-Seiten) verfügen, die eine Verbindung zu IBM WebSphere MQ herstellen, die für eine höhere Sicherheitsstufe als gewöhnlich konfiguriert sind.

Für IBM WebSphere MQ ist der Zugriff SYNCHRONIZE auf Anwendungsprozesse erforderlich, damit bestimmte Aktionen koordiniert werden können. In APAR IC35116 wurde IBM WebSphere MQ geändert, sodass die entsprechenden Berechtigungen angegeben werden. Für das Konto, unter dem IBM WebSphere MQ-Prozesse ausgeführt werden, sind eventuell aber zusätzliche Berechtigungen erforderlich, damit der angeforderte Zugriff erteilt werden kann.

Wenn eine Serveranwendung zum ersten Mal versucht, eine Verbindung zu einem Warteschlangenmanager herzustellen, ändert IBM WebSphere MQ den Prozess, um IBM WebSphere MQ -Administratoren die Berechtigung SYNCHRONIZE zu erteilen. Führen Sie die folgenden Schritte aus, um die Zusatzberechtigung für die Benutzer-ID zu konfigurieren, unter der IBM WebSphere MQ-Prozesse ausgeführt werden:

1. Starten Sie das Tool Local Security Policy, klicken Sie auf Security Settings-> Local Policies-> User Right Assignments, und klicken Sie auf "Debug Programs".
2. Doppelklicken Sie auf 'Debuggen von Programmen' und fügen Sie Ihre IBM WebSphere MQ-Benutzer-ID zur Liste hinzu.

Wenn sich das System in einer Windows -Domäne befindet und die effektive Richtlinieneinstellung immer noch nicht festgelegt ist, obwohl die lokale Richtlinieneinstellung festgelegt ist, muss die Benutzer-ID auf Domänenebene mit dem Tool für Domänensicherheitsrichtlinien auf dieselbe Weise berechtigt werden.

Sicherheit unter HP Integrity NonStop Server einrichten

Sicherheitsaspekte, die speziell für HP Integrity NonStop Server -Systeme gelten.

Der IBM WebSphere MQ-Client für HP Integrity NonStop Server unterstützt sowohl das TLS-Protokoll (Transport Layer Security) als auch das SSL-Protokoll (Secure Sockets Layer), um bei der Herstellung von Verbindungen zu einem Warteschlangenmanager Sicherheit auf Verbindungsebene bereitzustellen. Diese Protokolle werden unterstützt, indem eine Implementierung von OpenSSL verwendet wird. OpenSSL erfordert eine Quelle zufälliger Daten, um starke kryptografische Operationen bereitzustellen.

OpenSSL

OpenSSL -Sicherheitsübersicht für IBM WebSphere MQ -Client für HP Integrity NonStop Server.

Das OpenSSL-Toolkit ist eine Open-Source-Implementierung der Protokolle Secure Sockets Layer (SSL) und Transport Layer Security (TLS) für die sichere Kommunikation über ein Netz.

Das Toolkit wird vom OpenSSL-Projekt entwickelt. Weitere Informationen zum OpenSSL-Projekt finden Sie unter https://www.openssl.org. Der IBM WebSphere MQ-Client für HP Integrity NonStop Server enthält geänderte Versionen der OpenSSL-Bibliotheken und des Befehls **openssl**. Die Bibliotheken und der Befehl **openssl** werden aus dem OpenSSL -Toolkit 1.0.1cportiert und nur als Objektcode bereitgestellt. Es wird kein Quellcode bereitgestellt.

Die OpenSSL-Bibliotheken werden je nach Bedarf dynamisch von den IBM WebSphere MQ-Clientanwendungsprogrammen geladen. Nur die von IBM WebSphere MQ bereitgestellten OpenSSL-Bibliotheken werden für die Verwendung mit IBM WebSphere MQ-Clientanwendungen unterstützt.

Der Befehl **openssl** , der für die Zertifikatsverwaltung verwendet werden kann, ist im OSS-Verzeichnis `opt_installation_path/opt/mqm/bin`installiert.

Mit dem Befehl **openssl** können Sie Schlüssel und digitale Zertifikate mit verschiedenen gängigen Datenformaten erstellen, verwalten und einfache CA-Tasks ausführen.

Das Standardformat für Schlüssel-und Zertifikatsdaten, die von OpenSSL verarbeitet werden, ist das PEM-Format (Privacy Enhanced Mail). Daten im PEM-Format sind base64-codierte ASCII-Daten. Die Daten können somit über text-basierte Systeme wie E-Mail übertragen und mit Hilfe von Texteditoren und Webbrowsern ausgeschnitten und eingeteigt werden. PEM ist ein Internet-Standard für den textbasierten kryptografischen Austausch und wird in den Internet-RFCs 1421, 1422, 1423 und 1424 angegeben. IBM WebSphere MQ geht davon aus, dass eine Datei mit der Erweiterung `.pem` Daten im PEM-Format enthält. Eine Datei im PEM-Format kann mehrere Zertifikate und andere codierte Objekte enthalten und kann Kommentare enthalten.

Der IBM WebSphere MQ SSL-Support auf anderen Betriebssystemen kann die Verschlüsselung von Schlüssel- und Zertifikatdaten mithilfe der Distinguished Encoding Rules (DER) erforderlich machen. DER ist eine Gruppe von Codierungsregeln für die Verwendung der Notation ASN.1 in der sicheren Kommunikation. Daten, die mit DER codiert werden, sind binäre Daten, und das Format von Schlüssel-und Zertifikatdaten, die mit DER codiert werden, wird auch als PKCS#12 oder PFX bezeichnet. Eine Datei, die diese Daten enthält, hat im Allgemeinen eine Erweiterung von `.p12` oder `.pfx`. Der Befehl **openssl** kann zwischen dem PEM-Format und dem Format PKCS#12 konvertieren.

Entropiedämon

OpenSSL erfordert eine Quelle zufälliger Daten, um starke kryptografische Operationen bereitzustellen. Zufallszahlengenerierung ist eine Funktion, die normalerweise vom Betriebssystem oder von einem systemweiten Dämonprozess bereitgestellt wird. Das Betriebssystem von HP Integrity NonStop Server stellt diese Funktion innerhalb des Betriebssystems nicht bereit.

Bei der Verwendung des mit dem IBM WebSphere MQ-Client für HP Integrity NonStop Server bereitgestellten SSL- und TLS-Supports ist ein als Entropiedämon bezeichneter Prozess als Zufallsdatenquelle erforderlich. Beim Start eines Clientkanals, der SSL oder TLS erfordert, wird von OpenSSL ein aktiver Entropiedämon erwartet, der seine Services an einem Socket im OSS-Dateisystem unter `/etc/egd-pool` bereitstellt.

Vom IBM WebSphere MQ -Client für HP Integrity NonStop Serverwird kein Entropiedämon bereitgestellt. Der IBM WebSphere MQ -Client für HP Integrity NonStop Server wird mit den folgenden Entropiedämonen getestet:

- amqjkdm0 (wie vom IBM WebSphere MQ 5.3-Server bereitgestellt)
- `/usr/local/bin/prngd` (Version 0.9.27, wie von HP Integrity NonStop Server Open Source Technical Library bereitgestellt)

IBM WebSphere MQ MQI-Clientsicherheit einrichten

Sie müssen die IBM WebSphere MQ MQI-Clientsicherheit berücksichtigen, damit die Clientanwendungen keinen uneingeschränkten 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.

  Sie sollten die gegenseitige Authentifizierung in SSL oder TLS verwenden. Weitere Informationen hierzu finden Sie in „Mit SSL oder TLS arbeiten” auf Seite 117.
- 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. Bei beiden Funktionen besteht die Möglichkeit, die Benutzer-ID zu ändern, die für die Überprüfung der Zugriffsrechte verwendet wird.

  Weitere Informationen zur Zugriffssteuerung finden Sie unter „Autorisieren des Zugriffs auf Objekte” auf Seite 168.

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 „Bestätigte Client-Benutzer-ID einer MCAUSER-Benutzer-ID zuordnen” auf Seite 196.

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 „Kanalauthentifizierungsdatensätze” auf Seite 42.

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.

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 auf drei Arten festlegen, dass ein SSL- oder TLS-Kanal nur FIPS-zertifizierte CipherSpecs verwenden darf, die im Folgenden in ihrer Ausführungspriorität aufgeführt 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 aktuell über keine aktiven SSL- oder TLS-Verbindungen verfügt und ein gültiger FipsRequired-Wert in einer SSL-MQCONNX-Verbindung angegeben ist, dürfen alle nachfolgenden SSL-Verbindungen, die diesem Prozess zugeordnet sind, nur die diesem Wert zugeordneten CipherSpecs verwenden. Dies gilt, bis diese und alle anderen SSL- oder TLS-Verbindungen gestoppt wurden, woraufhin eine nachfolgende MQCONNX-Verbindung einen neuen Wert für FipsRequired bereitstellen kann.

Wenn Verschlüsselungshardware vorhanden ist, können die Verschlüsselungsmodule, die von WebSphere MQ verwendet werden, so konfiguriert werden, dass es sich um die vom Hardwareprodukt bereitgestellten Module handelt, die möglicherweise für eine bestimmte Stufe FIPS-zertifiziert sind. 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. WebSphere MQ garantiert nicht die Ablehnung aller Verbindungen dieses Typs und es liegt in der Verantwortlichkeiten des Kunden, festzustellen, ob Ihre WebSphere MQ-Konfiguration FIPS-kompatibel ist.

Zugehörige Konzepte

„Federal Information Processing Standards (FIPS) für UNIX, Linuxund Windows” auf Seite 27
Wenn die Verschlüsselung auf einem SSL-oder TLS-Kanal auf Windows-oder UNIX and Linux -Systemen erforderlich ist, verwendet WebSphere MQ ein Verschlüsselungspaket namens IBM Crypto for C (ICC). Auf den Plattformen Windows, UNIX and Linux , hat die ICC-Software das FIPS-Verschlüsselungsprogramm (Federal Information Processing Standards) Cryptomodule Validation Program des US National Institute of Standards and Technology auf Stufe 140-2 bestanden.

SSL-Zeilengruppe der Clientkonfigurationsdatei

Zugehörige Verweise

FipsRequired (MQLONG)

MQSSLFIPS

SSL- oder TLS-Clientanwendungen mit mehreren Installationen von GSKit V8.0 unter AIX ausführen

Bei unter AIX laufenden SSL- oder TLS-Clientanwendungen können die Fehler MQRC_CHANNEL_CONFIG_ERROR und AMQ6175 auftreten, wenn sie auf AIX-Systemen mit mehreren Installationen von GSKit V8.0 ausgeführt werden.

Beim Ausführen von Clientanwendungen auf einem AIX-System mit mehreren Installationen von GSKit V8.0 können Verbindungsaufrufe des Clients den Fehler MQRC_CHANNEL_CONFIG_ERROR zurückgeben, wenn SSL oder TLS verwendet wird. Der /var/mqm/errors protokolliert den Datensatzfehler AMQ6175 und AMQ9220 für die fehlgeschlagene Clientanwendung, z. B.:

```
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 oder LD_LIBRARY_PATH dazu geführt hat, dass derIBM WebSphere MQ Client einen gemischten Satz von Bibliotheken aus zwei verschiedenen GSKit V8.0-Installationen geladen hat. Die Ausführung einerIBM WebSphere MQ -Clientanwendung in einer Db2 -Umgebung kann diesen Fehler verursachen.

Um diesen Fehler zu vermeiden, schließen Sie die IBM WebSphere MQ -Bibliotheksverzeichnisse am Anfang des Bibliothekspfads ein, damit die IBM WebSphere 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 im Abschnitt setmqenv (WebSphere MQ -Umgebung festlegen) .

Kommunikation für SSL oder TLS auf UNIX, Linux, and Windows -Systemen 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 UNIX-, Linux -und Windows -Systemen können Sie die Tests mit selbst signierten Zertifikaten ausführen.

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 oder TLS auf UNIX, Linux, and Windows -Systemen arbeiten” auf Seite 120.

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 WebSphere MQ fordert der SSL- oder TLS-Server immer ein Zertifikat vom Client an.

Auf UNIX-, Linux- und Windows-Systemen sendet der SSL- oder TLS-Client ein Zertifikat nur dann, wenn es im richtigen IBM WebSphere 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 WebSphere MQ-Client ist dies `ibmwebspheremq` gefolgt von Ihrer Anmeldebenutzer-ID in Kleinschreibung, z. B. `ibmwebspheremqmyuserid`.

IBM WebSphere 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 im Abschnitt „Zwei Warteschlangenmanager über SSL oder TLS verbinden” auf Seite 218.

Mit SSL oder TLS arbeiten

In diesen Abschnitten finden Sie Anweisungen für die Ausführung von einzelnen Tasks im Zusammenhang mit der Verwendung von SSL oder TLS mit IBM WebSphere 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 153
- „Autorisieren des Zugriffs auf Objekte” auf Seite 168
- „Vertraulichkeit von Nachrichten” auf Seite 218
- „Datenintegrität von Nachrichten” auf Seite 240
- „Cluster sicher halten” auf Seite 259

Unter HP Integrity NonStop Server mit SSL oder TLS arbeiten

Beschreibt die OpenSSL-Sicherheitsimplementierung des IBM WebSphere MQ-Clients für HP Integrity NonStop Server einschließlich der Sicherheitsservices, der Komponenten, der unterstützten Protokollversionen, der unterstützten CipherSpecs und der nicht unterstützten Sicherheitsfunktionalität.

Die SSL-und TLS-Unterstützung von IBM WebSphere MQ bietet die folgenden Sicherheitsservices für Clientkanäle:

- Authentifizierung des Servers und, optional, Authentifizierung des Clients.
- Verschlüsseln und Entschlüsseln der Daten, die über einen Kanal fließen.
- Integritätsprüfungen für die Daten, die über einen Kanal fließen.

Die mit dem IBM WebSphere MQ-Client für HP Integrity NonStop Server bereitgestellte SSL- und TLS-Unterstützung umfasst die folgenden Komponenten:

- OpenSSL -Bibliotheken und der Befehl **`openssl`** .
- Den IBM WebSphere MQ-Kennwortstashbefehl **`amqrsslc`**.

Die folgenden für den SSL- oder TLS-Clientkanalbetrieb erforderlichen Komponenten werden nicht mit IBM WebSphere MQ-Client für HP Integrity NonStop Server bereitgestellt:

- Ein Entropy-Dämon, der eine Quelle für Zufallsdaten für die OpenSSL-Verschlüsselung bereitstellt.

Unterstützte Protokollversionen

Der IBM WebSphere MQ -Client für HP Integrity NonStop Server unterstützt die folgenden Protokollversionen:

- SSL 3.0
- TLS 1.0
- TLS 1.2

Unterstützte CipherSpecs

Der IBM WebSphere MQ -Client für HP Integrity NonStop Server unterstützt die folgenden CipherSpecs -Versionen:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- RC4_SHA_US
- RC4_MD5_US
- TRIPLE_DES_SHA_US
- TLS_RSA_WITH_3DES_EDE_CBC_SHA (nicht weiter unterstützt)
- DES_SHA_EXPORT1024
- RC4_56_SHA_EXPORT1024
- RC4_MD5_EXPORT
- RC2_MD5_EXPORT
- DES_SHA_EXPORT
- TLS_RSA_WITH_DES_CBC_SHA
- NULL_SHA
- NULL_MD5
- FIPS_WITH_DES_CBC_SHA
- FIPS_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_NULL_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- ECDHE_ECDSA_AES_128_CBC_SHA256
- ECDHE_ECDSA_AES_256_CBC_SHA384
- ECDHE_RSA_AES_128_CBC_SHA256
- ECDHE_RSA_AES_256_CBC_SHA384
- ECDHE_ECDSA_AES_128_GCM_SHA256
- ECDHE_ECDSA_AES_256_GCM_SHA384
- ECDHE_RSA_AES_128_GCM_SHA256
- ECDHE_RSA_AES_256_GCM_SHA384

Nicht unterstützte Sicherheitsfunktionalität

Der IBM WebSphere MQ -Client für HP Integrity NonStop Server unterstützt derzeit Folgendes nicht:

- PKCS#11 Cryptographic Hardware Support
- Überprüfung der Widerrufsliste für LDAP-Zertifikate
- Überprüfung des OCSP-Online-Zertifikatsprotokolls
- FIPS 140-2, NSA SUITE B Cipher Suite-Steuerelemente

Zertifikatsmanagement

Verwenden Sie eine Gruppe von Dateien zum Speichern von digitalen Zertifikats-und Zertifikatwiderrufinformationen.

Die IBM WebSphere MQ-SSL- und -TLS-Unterstützung verwendet eine Gruppe von Dateien zur Speicherung digitaler Zertifikate und Zertifikatswiderrufsinformationen. Diese Dateien befinden sich in einem Verzeichnis, das über das Feld "KeyRepository" in der MQSCO-Struktur angegeben ist, die über den MQCONNX-Aufruf übergeben wird, durch die Umgebungsvariable *MQSSLKEYR* oder in der SSL-Zeilengruppe des `mqclient.ini` mit dem Attribut "SSLKeyRepository".

Die MQSCO-Struktur hat Vorrang vor der Umgebungsvariablen MQSSLKEYR, die Vorrang vor dem Zeilengruppenwert INI-Datei hat.

Wichtig: Die Position des Schlüsselrepositorys gibt eine Verzeichnisposition und nicht einen Dateinamen auf der HP Integrity NonStop Server- Plattform an.

IBM WebSphere MQ-Client für HP Integrity NonStop Server verwendet am Speicherort des Schlüsselrepositorys die folgenden benannten Dateien (Groß-/Kleinschreibung beachten):

- „Persönlicher Zertifikatsspeicher” auf Seite 119
- „Truststore für Zertifikate” auf Seite 119
- „Ausdrucks-Stashdatei übergeben” auf Seite 120
- „Zertifikatswiderrufslistendatei” auf Seite 120

Persönlicher Zertifikatsspeicher
Die persönliche Zertifikatsspeicherdatei `cert.pem`.

Diese Datei enthält das persönliche Zertifikat und den verschlüsselten privaten Schlüssel für den Client, der im PEM-Format verwendet werden soll. Bei Verwendung von SSL- oder TLS-Kanälen, die keine Clientauthentifizierung erfordern, ist das Vorhandensein dieser Datei optional. Wenn die Clientauthentifizierung für den Kanal erforderlich ist, und SSLCAUTH (REQUIRED) in der Kanaldefinition angegeben ist, muss diese Datei vorhanden sein und sowohl das Zertifikat als auch den verschlüsselten privaten Schlüssel enthalten.

Die Dateiberechtigungen müssen in dieser Datei festgelegt werden, damit der Lesezugriff auf den Eigner des Zertifikatsspeichers möglich ist.

Eine ordnungsgemäß formatierte `cert.pem` -Datei muss genau zwei Abschnitte mit den folgenden Kopf- und Fußzeilen enthalten:

```
-----BEGIN PRIVATE KEY-----
Base 64 ASCII encoded private key data here
-----END PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
Base 64 ASCII encoded certificate data here
-----END CERTIFICATE-----
```

Der Verschlüsselungstext für den verschlüsselten privaten Schlüssel wird in der Stashdatei der Arbeitspassphrase `Stash.sth` gespeichert.

Truststore für Zertifikate
Die Truststore-Datei des Zertifikats `trust.pem`.

Diese Datei enthält die Zertifikate, die für die Validierung der persönlichen Zertifikate benötigt werden, die von Warteschlangenmanagern verwendet werden, zu denen der Client eine Verbindung herstellt, im PEM-Format. Der Zertifikatstruststore ist für alle SSL- oder TLS-Clientkanäle obligatorisch.

Die Dateiberechtigungen müssen so festgelegt werden, dass der Schreibzugriff auf diese Datei eingeschränkt wird.

Eine ordnungsgemäß formatierte `trust.pem` -Datei muss einen oder mehrere Abschnitte mit den folgenden Kopf-und Fußzeilen enthalten:

```
-----BEGIN CERTIFICATE-----
Base 64 ASCII encoded certificate data here
-----END CERTIFICATE-----
```

Ausdrucks-Stashdatei übergeben
Die Stashdatei der Arbeitspassphrase `Stash.sth`.

Diese Datei verfügt über ein internes IBM WebSphere MQ-Binärformat und enthält die verschlüsselte Kennphrase, die zum Zugriff auf den privaten Schlüssel in der Datei `cert.pem` erforderlich ist. Der private Schlüssel selbst wird im Zertifikatsspeicher von `cert.pem` gespeichert.

Diese Datei kann über das Befehlszeilentool IBM WebSphere MQ **amqrsslc** und den Parameter **-s** erstellt oder geändert werden. Beispiel: Wenn das Verzeichnis `/home/alice` eine `cert.pem` -Datei enthält:

```
amqrsslc -s /home/alice/cert

  Enter password for Keystore /home/alice/cert.pem :
    password

  Stashed the password in file /home/alice/Stash.sth
```

Die Dateiberechtigungen müssen in dieser Datei festgelegt werden, damit der Lesezugriff auf den Eigner des zugeordneten persönlichen Zertifikatsspeichers möglich ist.

Zertifikatswiderrufslistendatei
Die Zertifikatswiderrufslistendatei `crl.pem`.

Diese Datei enthält die Zertifikatswiderrufslisten (Certificate Revocation Lists, CRLs), die der Client verwendet, um digitale Zertifikate im PEM-Format zu validieren. Das Vorhandensein dieser Datei ist optional. Wenn diese Datei nicht vorhanden ist, werden keine Zertifikatswiderrufsprüfungen durchgeführt, wenn Zertifikate geprüft werden.

Die Dateiberechtigungen müssen so festgelegt werden, dass der Schreibzugriff auf diese Datei eingeschränkt wird.

Eine ordnungsgemäß formatierte `crl.pem` -Datei muss einen oder mehrere Abschnitte mit den folgenden Kopf-und Fußzeilen enthalten:

```
-----BEGIN X509 CRL-----
Base 64 ASCII encoded CRL data here
-----END X509 CRL-----
```

Mit SSL oder TLS auf UNIX, Linux, and Windows -Systemen arbeiten

Auf UNIX-, Linux -und Windows -Systemen wird die SSL-Unterstützung (Secure Sockets Layer) mit IBM WebSphere MQinstalliert.

Weitere Informationen zu Zertifikatsprüfungs-Richtlinien finden Sie im Abschnitt Certificate Validation and Trust Policy Design .

iKeyman, iKeycmd, runmqakm und runmqckm verwenden

Auf UNIX-, Linux -und Windows -Systemen können Sie Schlüssel und digitale Zertifikate über die iKeyman -GUI oder über die Befehlszeile mit iKeycmd oder runmqakm verwalten.

- Für **UNIX and Linux** -Systeme:
  - Verwenden Sie den Befehl **strmqikm** , um die iKeyman -GUI zu starten.
  - Mit dem Befehl **runmqckm** können Sie Tasks mit der Befehlszeilenschnittstelle iKeycmd ausführen.
  - Verwenden Sie den Befehl **runmqakm** , um Tasks mit der Befehlszeilenschnittstelle 'runmqakm' auszuführen. Die Befehlssyntax für **runmqakm** entspricht der Syntax für **runmqckm**.

    Wenn Sie SSL-Zertifikate auf FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** statt der Befehle **runmqckm** und **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, muss es sich bei iKeycmd und iKeyman um 64-Bit-Programme handeln. 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 von Windows und Linux x86 sind die einzigen Ausnahmen, da die Programme iKeyman und iKeycmd auf diesen Plattformen 32-Bit sind.

  Auf den folgenden Plattformen, auf denen die JRE in früheren Versionen des Produkts 32 Bit war, aber nur in IBM WebSphere MQ Version 7.564 Bit ist, müssen Sie möglicherweise zusätzliche PKCS#11 -Treiber installieren, die für den Adressierungsmodus der **iKeyman** -und **iKeycmd** JRE geeignet sind. da der PKCS#11-Treiber denselben Adressierungsmodus wie die JRE verwenden muss. In der folgenden Tabelle sind die JRE-Adressmodi in IBM WebSphere MQ Version 7.5 aufgeführt.

Tabelle 12. IBM WebSphere MQ Version 7.5 JRE-Adressierungsmodi

Plattform	**JRE-Adressmodus**
Windows (32-Bit oder 64-Bit)	32
Linux für System x 32 Bit	32
Linux for System x 64-Bit	64
Linux für System p	64
Linux für System z	64
HP-UX	64
Solaris SPARC	64
Solaris x86-64	64
AIX	64

  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
    ```

- Für **Windows** -Systeme:
  - Verwenden Sie den Befehl **strmqikm** , um die iKeyman -GUI zu starten.
  - Mit dem Befehl **runmqckm** können Sie Tasks mit der Befehlszeilenschnittstelle iKeycmd ausführen.

Wenn Sie SSL-Zertifikate auf FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** statt der Befehle **runmqckm** und **strmqikm**.

Informationen zum Anfordern der SSL-Traceerstellung auf Systemen mit UNIX, Linux oder Windows finden Sie unter strmqtrc.

Zugehörige Verweise

runmqckm-und runmqakm-Befehle

Schlüsselrepository auf Systemen mit UNIX, Linux, and Windows einrichten

Sie können ein Schlüsselrepository über die Benutzerschnittstelle von iKeyman oder über die Befehle **iKeycmd** oder **runmqakm** einrichten.

Informationen zu diesem Vorgang

Für eine SSL- oder TLS-Verbindung muss an jedem Ende der Verbindung ein *Schlüsselrepository* vorhanden sein. Jeder IBM WebSphere MQ -Warteschlangenmanager und IBM WebSphere MQ MQI client muss Zugriff auf ein Schlüsselrepository haben. Weitere Informationen finden Sie in „Das SSL- oder TLS-Schlüsselrepository” auf Seite 25.

Auf UNIX, Linux, and Windows -Systemen werden digitale Zertifikate in einer Schlüsseldatenbankdatei gespeichert, die über die **iKeyman** -Benutzerschnittstelle oder mit den Befehlen **iKeycmd** oder **runmqakm** verwaltet wird. Diese digitalen Zertifikate weisen Beschriftungen auf. Ein bestimmter Kennsatz verknüpft ein persönliches Zertifikat mit einem Warteschlangenmanager oder IBM WebSphere MQ MQI client. SSL und TLS verwenden dieses Zertifikat zu Authentifizierungszwecken. Auf UNIX, Linux, and Windows -Systemen verwendet IBM WebSphere MQ `ibmwebspheremq` als Kennsatzpräfix, um Verwechslungen mit Zertifikaten für andere Produkte zu verhindern. Auf das Präfix folgt der Name des Warteschlangenmanagers oder der Anmelde-ID des IBM WebSphere MQ MQI client -Benutzers, der in Kleinbuchstaben geändert wird. Stellen Sie sicher, dass Sie die gesamte Zertifikatsbezeichnung in Kleinbuchstaben angeben.

Der Name der Schlüsseldatenbankdatei enthält einen Pfad und einen Stammnamen:

- Auf UNIX and Linux-Systemen lautet der Standardpfad für einen Warteschlangenmanager (der beim Erstellen des Warteschlangenmanagers festgelegt wird) folgendermaßen: `/var/mqm/qmgrs/<queue_manager_name>/ssl`.

  Auf Windows -Systemen ist der Standardpfad *MQ_INSTALLATION_PATH*`\Qmgrs\`*queue_manager_name*`\ssl`, wobei *MQ_INSTALLATION_PATH* das Verzeichnis ist, in dem IBM WebSphere MQ installiert ist. Beispiel: `C:\program files\IBM\WebSphere MQ\Qmgrs\QM1\ssl`.

  Der Standardstammname ist `key`. Wahlweise können Sie einen 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 WebSphere MQ -Client gibt es keinen Standardpfad oder Standardstammnamen. Der Zugriff auf diese Datei wird direkt gesteuert. Die Erweiterung muss `.kdb` sein.

Erstellen Sie keine Schlüsselrepositorys in einem Dateisystem, das Dateiebenensperren nicht unterstützt, z. B. NFS Version 2 auf Linux -Systemen.

Informationen zum Überprüfen und Angeben des Namens der Schlüsseldatenbankdatei finden Sie in „Position des Schlüsselrepositorys für einen Warteschlangenmanager auf UNIX-, Linux -oder Windows -Systemen ändern” auf Seite 127 . Sie können den Namen der Schlüsseldatenbankdatei entweder vor oder nach der Erstellung der Schlüsseldatenbankdatei angeben.

Die Benutzer-ID, unter der Sie den Befehl **iKeyman** oder **iKeycmd** ausführen, muss über Schreibberechtigung für das Verzeichnis verfügen, in dem die Schlüsseldatenbankdatei erstellt oder aktualisiert wird. Bei einem Warteschlangenmanager mit dem Standardverzeichnis `ssl` muss die Benutzer-ID, unter der Sie **iKeyman** oder **iKeycmd** ausführen, Mitglied der Gruppe 'mqm' sein. Wenn Sie für eine IBM WebSphere MQ MQI client-Instanz **iKeyman** oder **iKeycmd** mit einer anderen Benutzer-ID als der ausführen, unter der der Client ausgeführt wird, müssen Sie die Dateiberechtigungen ändern, damit IBM WebSphere MQ

MQI client zur Laufzeit auf die Schlüsseldatenbankdatei zugreifen kann. Weitere Informationen finden Sie in den Abschnitten „Auf Schlüsseldatenbankdateien unter Windows zugreifen und diese schützen” auf Seite 124 und „Zugriff auf Schlüsseldatenbankdateien auf UNIX and Linux -Systemen und Sichern der Schlüsseldatenbankdateien” auf Seite 125.

In **iKeyman** oder **iKeycmd** Version 7.0werden neue Schlüsseldatenbanken automatisch mit einer Gruppe vordefinierter Zertifikate einer Zertifizierungsstelle gefüllt. In **iKeyman** oder **iKeycmd** Version 8.0werden Schlüsseldatenbanken nicht automatisch gefüllt, wodurch die Erstkonfiguration sicherer wird, da Sie nur die gewünschten CA-Zertifikate in Ihre Schlüsseldatenbankdatei einschließen.

Anmerkung: Aufgrund dieser Verhaltensänderung für GSKit Version 8.0 , die dazu führt, dass CA-Zertifikate nicht mehr automatisch zum Repository hinzugefügt werden, müssen Sie Ihre bevorzugten CA-Zertifikate manuell hinzufügen. Diese Verhaltensänderung bietet Ihnen eine differenzierte Kontrolle über die verwendeten CA-Zertifikate. Siehe „Hinzufügen von Standard-CA-Zertifikaten zu einem leeren Schlüsselrepository auf UNIX, Linux, and Windows -Systemen mit GSKit Version 8.0” auf Seite 125.

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:
   - Auf UNIX, Linux, and Windows -Systemen:

     ```
     runmqckm -keydb -create -db filename -pw password -type cms -stash
     ```

   - runmqakm wird verwendet:

     ```
     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 WebSphere MQmuss dies cmssein.)

   -stash
   Speichert das Kennwort der Schlüsseldatenbank in einer Datei.

   -fips
   Inaktiviert die Verwendung der BSafe-Verschlüsselungsbibliothek. Im FIPS-Modus wird nur die ICC-Komponente verwendet, die für diesen Modus erfolgreich initialisiert werden muss. 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 befinden sich im standardmäßigen druckbaren ASCII-Zeichensatz im Bereich von 0x20 bis 0x7E.

Alternativ können Sie eine Schlüsseldatenbank über die Benutzerschnittstelle von **strmqikm** (iKeyman) erstellen.

2. Melden Sie sich auf UNIX 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 von iKeyman , 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` ist, 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 Warteschlangenmanager: `/var/mqm/qmgrs/QM1/ssl` (auf UNIX and Linux -Systemen) oder `C:\Program Files\IBM\WebSphere MQ\qmgrs\QM1\ssl` (auf Windows -Systemen)

     Der Pfad muss mit dem Wert des **SSLKeyRepository** -Attributs des Warteschlangenmanagers übereinstimmen.
   - Für einen IBM WebSphere MQ -Client: `/var/mqm/ssl` (auf UNIX and Linux -Systemen) oder `C:\mqm\ssl` (auf Windows -Systemen)
8. Klicken Sie auf **Öffnen** .

   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 verdeckt speichern, schlagen alle Versuche zum Starten von SSL- oder TLS-Kanälen fehl, da das Kennwort nicht für den Zugriff auf die Schlüsseldatenbankdatei abgerufen werden kann.
11. Klicken Sie auf **OK**.

    Das Fenster Personal Certificates wird geöffnet.
12. Legen Sie die Zugriffsberechtigungen wie unter „Auf Schlüsseldatenbankdateien unter Windows zugreifen und diese schützen” auf Seite 124 oder „Zugriff auf Schlüsseldatenbankdateien auf UNIX and Linux -Systemen und Sichern der Schlüsseldatenbankdateien” auf Seite 125 beschrieben fest.

Auf Schlüsseldatenbankdateien unter Windows 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 auf die Dateien *key*`.kdb`, *key*`.sth`, *key*`.crl` und *key*`.rdb` fest, wobei *key* für den Stammnamen Ihrer Schlüsseldatenbank steht, um eine Berechtigung für eine eingeschränkte Gruppe von Benutzern 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.

Zugriff auf Schlüsseldatenbankdateien auf UNIX and Linux -Systemen und Sichern der Schlüsseldatenbankdateien
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*`.crl` und *key*`.rdb` fest. Dabei ist *key* der Stammname Ihrer Schlüsseldatenbank, read und write für den Dateieigner und read für die mqm-oder Clientbenutzergruppe (-rw-r -----).

Hinzufügen von Standard-CA-Zertifikaten zu einem leeren Schlüsselrepository auf UNIX, Linux, and Windows -Systemen mit GSKit Version 8.0
Gehen Sie wie folgt vor, um ein oder mehrere der Standard-CA-Zertifikate einem leeren Schlüsselrepository mit GSKit Version 8 hinzuzufügen.

In GSKit Version 7.0bestand das Verhalten beim Erstellen eines neuen Schlüsselrepositorys darin, automatisch eine Gruppe von Standard-CA-Zertifikaten für häufig verwendete Zertifizierungsstellen hinzuzufügen. Bei GSKit Version 8 hat sich dieses Verhalten geändert, sodass dem Repository keine CA-Zertifikate mehr automatisch hinzugefügt werden. Der Benutzer muss jetzt CA-Zertifikate manuell in das Schlüsselrepository hinzufügen.

iKeyman verwenden

Führen Sie die folgenden Schritte auf der Maschine aus, auf der Sie das CA-Zertifikat hinzufügen möchten:

1. Starten Sie die grafische Benutzerschnittstelle iKeyman mit dem Befehl **`strmqikm`** (auf UNIX-, Linux -und Windows -Systemen).
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** . 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 auf der höchsten Ebene für die Organisation aus, deren Zertifikaten von Zertifizierungsstellen Sie vertrauen wollen, um die vollständige Liste mit gültigen Zertifikaten von Zertifizierungsstellen anzuzeigen.
11. Wählen Sie die Zertifikate von Zertifizierungsstellen, denen Sie vertrauen wollen, aus der Liste aus 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 mit iKeycmd 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 WebSphere MQ allen persönlichen Zertifikaten vertraut, die von diesem CA-Zertifikat signiert wurden. Überlegen Sie daher sorgfältig, welchen Zertifizierungsstellen Sie vertrauen wollen, und fügen Sie nur die Zertifikate von Zertifizierungsstellen hinzu, die Sie zur Authentifizierung Ihrer Clients und Manager benötigen. Es wird nicht empfohlen, die vollständige Gruppe von Standardzertifikaten von CA-Zertifikaten hinzuzufügen, es sei denn, dies ist eine definitive Voraussetzung für Ihre Sicherheitsrichtlinie.

Schlüsselrepository für einen Warteschlangenmanager auf UNIX, Linux, and Windows -Systemen suchen

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 WebSphere MQ-Explorer oder mit PCF-Befehlen anzeigen.
2. Untersuchen Sie die Befehlsausgabe für den Pfad und den Stammnamen der Schlüsseldatenbankdatei.

Zum Beispiel:

a. auf UNIX and Linux-Systemen: `/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`, wobei *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl` der Pfad und `key` der Stammname ist. *MQ_INSTALLATION_PATH* steht für das übergeordnete Verzeichnis, in dem WebSphere MQ installiert ist.

Position des Schlüsselrepositorys für einen Warteschlangenmanager auf UNIX-, Linux -oder Windows -Systemen 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 auf UNIX and Linux -Systemen:

```
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:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\ssl\Mykey')
```

Der vollständig qualifizierte Dateiname der Schlüsseldatenbankdatei lautet `C:\Program Files\IBM\WebSphere 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 die Attribute Ihres Warteschlangenmanagers auch mit den Befehlen WebSphere MQ Explorer oder PCF 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 erforderlichen CA-und persönlichen Zertifikaten füllen, wie in „Persönliches Zertifikat in ein Schlüsselrepository auf UNIX, Linux, and Windows -Systemen importieren” auf Seite 141beschrieben.

Schlüsselrepository für einen IBM WebSphere MQ MQI-Client auf UNIX, Linux, and Windows -Systemen suchen

Die Position des Schlüsselrepositorys wird durch die Variable MQSSLKEYR angegeben oder im MQCONNX-Aufruf angegeben.

Überprüfen Sie die Umgebungsvariable MQSSLKEYR, um die Speicherposition der Schlüsseldatenbankdatei Ihres IBM WebSphere MQ MQI-Clients abzurufen. Beispiel:

```
echo $MQSSLKEYR
```

Überprüfen Sie auch Ihre Anwendung, da der Name der Schlüsseldatenbankdatei auch in einem MQCONNX-Aufruf festgelegt werden kann (siehe „Schlüsselrepositoryposition für einen IBM WebSphere MQ MQI-Client auf UNIX, Linux, and Windows -Systemen angeben” auf Seite 128). Der in einem MQCONNX-Aufruf festgelegte Wert überschreibt den Wert von MQSSLKEYR.

Schlüsselrepositoryposition für einen IBM WebSphere MQ MQI-Client auf UNIX, Linux, and Windows -Systemen angeben

Es gibt kein Standardschlüsselrepository für einen IBM WebSphere MQ MQI-Client. 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 haben zwei Möglichkeiten, die Position der Schlüsseldatenbankdatei Ihres IBM WebSphere MQ MQI-Clients anzugeben:

- Definieren Sie die Umgebungsvariable MQSSLKEYR. Beispiel auf UNIX and Linux -Systemen:

  ```
  export MQSSLKEYR=/var/mqm/ssl/key
  ```

  Die Schlüsseldatenbankdatei hat den vollständig qualifizierten Dateinamen:

  ```
  /var/mqm/ssl/key.kdb
  ```

  Unter Windows:

  ```
  set MQSSLKEYR=C:\Program Files\IBM\WebSphere MQ\ssl\key
  ```

  Die Schlüsseldatenbankdatei hat den vollständig qualifizierten Dateinamen:

  ```
  C:\Program Files\IBM\WebSphere 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 im Abschnitt Übersicht für MQSCO .

Wenn Änderungen an Zertifikaten oder dem Zertifikatsspeicher auf UNIX-, Linux -oder Windows-Systemen 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 einzelner, abgehender Kanalprozess zum ersten Mal einen SSL-Kanal ausführt.
- Wenn ein neuer einzelner, ankommender TCP/IP-Kanalprozess zum ersten Mal eine Anforderung zum Start eines SSL-Kanals empfängt.
- Wenn die WebSphere MQ SSL-Umgebung mit dem MQSC-Befehl REFRESH SECURITY TYPE(SSL) aktualisiert wird.
- Bei Clientanwendungsprozessen wenn die letzte SSL-Verbindung in dem Prozess geschlossen wird. Bei der nächsten SSL-Verbindung werden die Zertifikatsänderungen berücksichtigt.
- Bei Kanälen, die als Threads eines Prozesses für Prozess-Pooling (amqrmppa) ausgeführt werden: Wenn der Prozess für das Prozess-Pooling gestartet bzw. erneut gestartet wird und zuerst einen SSL-Kanal ausführt. Wenn der Prozess für das Prozess-Pooling bereits einen SSL-Kanal ausgeführt hat und die Änderung sofort wirksam werden soll, führen sie den MQSC-Befehl REFRESH SECURITY TYPE(SSL) aus.
- Bei Kanälen, die als Threads des Kanalinitiators ausgeführt werden: Wenn der Kanalinitiator gestartet bzw. erneut gestartet wird und zuerst einen SSL-Kanal aktiviert. Wenn der Kanalinitiatorprozess bereits einen SSL-Kanal ausgeführt hat und die Änderung sofort wirksam werden soll, führen sie den MQSC-Befehl REFRESH SECURITY TYPE(SSL) aus.

- Bei Kanälen, die als Threads eines TCP/IP-Empfangsprogramms ausgeführt werden: Wenn das Empfangsprogramm gestartet bzw. erneut gestartet wird und zum ersten Mal eine Anforderung empfängt, einen SSL-Kanal zu starten. Wenn das Empfangsprogramm bereits einen SSL-Kanal ausgeführt hat und die Änderung sofort wirksam werden soll, führen sie den MQSC-Befehl REFRESH SECURITY TYPE(SSL) aus.

Sie können die SSL-Umgebung von WebSphere MQ auch mit den IBM WebSphere MQ Explorer-oder PCF-Befehlen aktualisieren.

Selbst signiertes persönliches Zertifikat auf Systemen mit UNIX, Linux, and Windows erstellen

Sie können ein selbst signiertes Zertifikat mit iKeyman, iKeycmdoder runmqakm erstellen.

Anmerkung: IBM WebSphere 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 dazu, warum Sie selbst signierte Zertifikate verwenden sollten, finden Sie unter „Selbst signierte Zertifikate für die gegenseitige Authentifizierung zweier Warteschlangenmanager verwenden” auf Seite 219.

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. WebSphere MQ unterstützt drei verschiedene Typen von CipherSpec. Weitere Informationen finden Sie unter „Interoperabilität von Elliptic Curve und RSA CipherSpecs” auf Seite 37 im Abschnitt „Digitale Zertifikate und CipherSpec -Kompatibilität in IBM WebSphere MQ” auf Seite 36 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 .

iKeyman verwenden

iKeyman stellt keine FIPS-kompatible Option bereit. Wenn Sie SSL-oder TLS-Zertifikate auf FIPS-konforme Weise verwalten müssen, verwenden Sie den Befehl **runmqakm** .

Gehen Sie wie folgt vor, um ein selbst signiertes Zertifikat für Ihren WS-Manager oder WebSphere MQ MQI-Client abzurufen:

1. Starten Sie die GUI iKeyman 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, in der das Zertifikat gespeichert 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. 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 im Feld **Schlüsselkennsatz** Folgendes ein:
   - Für einen Warteschlangenmanager ibmwebspheremq , gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben. Beispiel: Für QM1, ibmwebspheremqqm1oder

- Bei einem WebSphere MQ -Client `ibmwebspheremq` gefolgt von Ihrer Anmeldebenutzer-ID in Kleinbuchstaben, z. B. `ibmwebspheremqmyuserid` .

10. Geben Sie einen Wert für ein beliebiges Feld in der **Distinguished name**oder für eines der **Subject alternative name** -Felder ein oder wählen Sie einen Wert 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 11.
12. Klicken Sie auf **OK**. In der Liste **Persönliche Zertifikate** wird die Bezeichnung des selbst signierten persönlichen Zertifikats angezeigt, das Sie erstellt haben.

Verwenden der Befehlszeile

Verwenden Sie die folgenden Befehle, um ein selbst signiertes persönliches Zertifikat mit iKeycmd oder runmqakm zu erstellen:

- iKeycmd auf UNIX-, Linux -und Windows -Systemen:

```
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_dsname` *`DNS_names`* , `-san_email-addr` *`email_addresses`* oder `-san_ipaddr` *`IP_addresses`* verwenden.

- runmqakm wird verwendet:

```
runmqakm -cert -create -db filename -pw
password -label label
        -dn distinguished_name -size key_size
 -x509version version -expire days

        -fips -sig_alg algorithm
```

`-db` *`filename`*	Der vollständig qualifizierte Dateiname einer CMS-Schlüsseldatenbank.
`-pw` *`password`*	Das Kennwort für die CMS-Schlüsseldatenbank.
`-label` *`label`*	Der Schlüsselkennsatz, der dem Zertifikat zugeordnet ist.
`-dn` *`distinguished_name`*	Der definierte X.500-Name ist in doppelte Anführungszeichen eingeschlossen. Mindestens ein Attribut ist erforderlich. Sie können mehrere OU-oder DC-Attribute angeben.
`-size` *`key_size`*	Die Schlüsselgröße. Für iKeycmdkann der Wert 512 oder 1024 sein. Für runmqakm kann der Wert 512, 1024, 2048 oder 4096 angegeben werden.
`-x509version` *`version`*	Die Version des zu erstellenden X.509-Zertifikats. Der Wert kann 1, 2 oder 3 sein. Der Standardwert ist 3.
`-expire` *`days`*	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. Dieser Modus deaktiviert die Verwendung der BSafe-Verschlüsselungsbibliothek. Im FIPS-Modus wird nur die ICC-Komponente verwendet, die für diesen Modus erfolgreich initialisiert werden muss. 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`	Für runmqakm der Hash-Algorithmus, der bei der Erstellung eines selbst signierten Zertifikats verwendet wird. Dieser Hashing-Algorithmus wird verwendet, um die Signatur zu erstellen, die dem neu erstellten selbst signierten Zertifikat zugeordnet ist. Mögliche Werte: `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` .
`-sig_alg`	Für iKeycmd der asymmetrische Signaturalgorithmus, der für die Erstellung des Schlüsselpaars des Eintrags verwendet wird. Gültige Werte: `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`oder `SHAWithRSA`. Der Standardwert ist `SHA1WithRSA` .
`-san_dnsname` *`DNS_names`*	Eine durch Kommas oder Leerzeichen getrennte Liste mit DNS-Namen für den Eintrag, der erstellt wird.
`-san_emailaddr` *`email_addresses`*	Eine durch Kommas oder Leerzeichen getrennte Liste der E-Mail-Adressen für den zu erstellenden Eintrag.
`-san_ipaddr` *`IP_addresses`*	Eine durch Kommas oder Leerzeichen getrennte Liste der IP-Adressen für den Eintrag, der erstellt wird.

distributed

Persönliches Zertifikat unter UNIX, Linux, and Windows anfordern

Sie können ein persönliches Zertifikat über die **strmqikm** anfordern (iKeyman) GUI oder über die Befehlszeile mit den Befehlen **runmqckm** oder **runmqakm** . 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 (GUI) von iKeyman oder über die Befehlszeile anfordern, wobei Sie die folgenden Hinweise beachten:

- WebSphere 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. WebSphere MQ unterstützt drei verschieden Typen von CipherSpec. Weitere Informationen finden Sie unter „Interoperabilität von Elliptic Curve und RSA CipherSpecs” auf Seite 37 im Abschnitt „Digitale Zertifikate und CipherSpec -Kompatibilität in IBM WebSphere MQ” auf Seite 36
- 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`.

- Nur der Befehl runmqakm stellt eine FIPS-kompatible Option bereit.
- Wenn Sie Verschlüsselungshardware verwenden, lesen Sie den Abschnitt „Anfordern eines persönlichen Zertifikats für Ihre PKCS #11-Hardware” auf Seite 149.

iKeyman-Benutzerschnittstelle verwenden

Informationen zu diesem Vorgang

iKeyman stellt keine FIPS-kompatible Option bereit. Wenn Sie SSL-oder TLS-Zertifikate auf 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 iKeyman-Benutzerschnittstelle verwenden:

1. Starten Sie die Benutzerschnittstelle iKeyman 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 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 im Feld **Schlüsselkennsatz** die folgenden Kennsätze ein:
   - Geben Sie für einen Warteschlangenmanager `ibmwebspheremq` gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben ein. Geben Sie für einen Warteschlangenmanager mit dem Namen `QM1`beispielsweise `ibmwebspheremqqm1`ein.
   - Bei einem IBM WebSphere MQ MQI clientgeben Sie `ibmwebspheremq` gefolgt von Ihrer Anmelde-Benutzer-ID in Kleinbuchstaben ein, z. B. `ibmwebspheremqmyuserid` .
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 11.
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 Datei gespeichert, die Sie in Schritt „11” auf Seite 132 ausgewählt haben.
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.

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 -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_email-addr *email_addresses* oder -san_ipaddr *IP_addresses* verwenden.

- runmqakm wird verwendet:

```
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.

-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.

-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. Dieser Modus deaktiviert die Verwendung der BSafe-Verschlüsselungsbibliothek. Im FIPS-Modus wird nur die ICC-Komponente verwendet, die für diesen Modus erfolgreich initialisiert werden muss. 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. Mögliche Werte sind MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA , SHA1WithRSA, SHA256_WITH_RSA, SHA256WithR-SA, SHA2WithRSA, SHA384_WITH_RSA , SHA384WithRSA, SHA512_WITH_RSA , SHA512WithR-SA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSAoder 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. Mögliche Werte: md5, MD5_WITH_RSA, MD5WithR-

SA, 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 .

-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.

Verlängern eines bestehenden persönlichen Zertifikats auf UNIX, Linux, and Windows-Systemen

Sie können ein persönliches Zertifikat über die Benutzerschnittstelle von iKeyman oder mit dem Befehl **iKeycmd** oder **runmqakm** verlängern.

Vorbereitende Schritte

Wenn Sie größere Schlüssel für Ihre persönlichen Zertifikate verwenden müssen, funktionieren die unten beschriebenen Erneuerungsschritte nicht, da die neu erstellte Zertifikatsanforderung aus einem vorhandenen Schlüssel generiert wird.

Führen Sie die in „Persönliches Zertifikat unter UNIX, Linux, and Windows anfordern” auf Seite 131 beschriebenen Schritte aus, um eine neue Zertifikatsanforderung unter Verwendung der erforderlichen Schlüsselgrößen zu erstellen. Dieser Prozess ersetzt Ihren vorhandenen Schlüssel.

Informationen zu diesem Vorgang

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.

iKeyman-Benutzerschnittstelle verwenden

Informationen zu diesem Vorgang

iKeyman stellt keine FIPS-kompatible Option bereit. Wenn Sie SSL-oder TLS-Zertifikate auf 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 iKeyman-Benutzerschnittstelle verwenden:

1. Starten Sie die Benutzerschnittstelle von iKeyman mit dem Befehl **strmqikm** auf UNIX, Linux, and Windows -Systemen.
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. Übernehmen Sie im Feld **Dateiname**entweder den Standardwert `certreq.arm`oder geben Sie einen neuen Wert einschließlich des vollständigen Dateipfads ein.
11. Klicken Sie auf **OK**. Die Zertifikatsanforderung wird in der Datei gespeichert, die Sie in Schritt „9” auf Seite 135 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 mit dem Befehl **iKeycmd** oder **runmqakm** anzufordern:

- **iKeycmd** auf UNIX, Linux, and Windows -Systemen verwenden:

```
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 auf UNIX-, Linux -und Windows -Systemen empfangen” auf Seite 136 beschriebenen Schritte zu Ihrer Schlüsseldatenbank hinzufügen.

Persönliche Zertifikate in einem Schlüsselrepository auf UNIX-, Linux -und Windows -Systemen 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.

iKeyman verwenden

Wenn Sie SSL-Zertifikate FIPS-konform verwalten müssen, verwenden Sie den Befehl 'runmqakm'. iKeyman stellt keine FIPS-kompatible Option bereit.

Stellen Sie sicher, dass die zu importierende Zertifikatsdatei Schreibberechtigung für den aktuellen Benutzer hat, und verwenden Sie dann die folgende Prozedur für einen Warteschlangenmanager oder einen WebSphere MQ MQI-Client, um ein persönliches Zertifikat in der Schlüsseldatenbankdatei zu empfangen:

1. Starten Sie die iKeyman -GUI mit dem Befehl **strmqikm** (unter Windows UNIX and Linux).
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 die folgenden Befehle, um ein persönliches Zertifikat mit iKeycmd

- Geben Sie unter UNIX, Linux und Windowsden folgenden Befehl aus:

```
runmqckm -cert -receive -file filename -db filename -pw
password
        -format ascii
```

Dabei gilt:

`-file` *filename*	ist der vollständig qualifizierte Dateiname der Datei, die das persönliche Zertifikat enthält.
`-db` *filename*	ist der vollständig qualifizierte Dateiname einer CMS-Schlüsseldatenbank.
`-pw` *password*	ist das Kennwort für die CMS-Schlüsseldatenbank.
`-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` .

Wenn Sie Verschlüsselungshardware verwenden, lesen Sie die Informationen in „Persönlichem Zertifikat in Ihre PKCS #11-Hardware importieren” auf Seite 151.

CA-Zertifikate aus einem Schlüsselrepository extrahieren

Gehen Sie wie folgt vor, um ein CA-Zertifikat zu extrahieren.

iKeyman verwenden

Wenn Sie SSL-Zertifikate FIPS-konform verwalten müssen, verwenden Sie den Befehl 'runmqakm'. 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 iKeyman -GUI 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-Zertifikat mit iKeycmd zu extrahieren:

- Unter UNIX, Linux und Windows:

```
runmqckm -cert -extract -db filename -pw password -label label -target filename
         -format ascii
```

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`.

Öffentlichen Teil eines selbst signierten Zertifikats aus einem Schlüsselrepository auf UNIX-, Linux -und Windows -Systemen extrahieren

Führen Sie die folgende Prozedur aus, um den öffentlichen Teil eines selbst signierten Zertifikats zu extrahieren.

iKeyman verwenden

Wenn Sie SSL-Zertifikate FIPS-konform verwalten müssen, verwenden Sie den Befehl 'runmqakm'. 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 iKeyman -GUI mit dem Befehl **strmqikm** (unter UNIX, Linux und 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 das Zertifikat extrahiert 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.
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 iKeycmd oder runmqakm zu extrahieren:

- Unter UNIX, Linux und Windows:

```
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`.

CA-Zertifikat (oder den öffentlichen Teil eines selbst signierten Zertifikats) in einem Schlüsselrepository auf UNIX, Linux, and Windows -Systemen 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: Ist das Zertifikat, das hinzugefügt werden soll, in einer Zertifikatskette enthalten, müssen Sie alle Zertifikate oberhalb dieses Zertifikats in der Kette ebenfalls hinzufügen. Das Zertifikat muss im ASCII-Format (UTF-8) oder im Binärformat (DER) vorliegen, da IBM Global Secure Toolkit (GSKit) keine andere Zertifikatscodierung unterstützt. Die Zertifikate müssen in abfolgender Reihenfolge hinzugefügt werden, ab dem Stammzertifikat, gefolgt von dem unmittelbar darunterliegenen CA-Zertifikat in der Kette.

iKeyman verwenden

Wenn Sie SSL-Zertifikate FIPS-konform verwalten müssen, verwenden Sie den Befehl 'runmqakm'. iKeyman 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 iKeyman mit dem Befehl **strmqikm** (auf UNIX-, Linux -und Windows -Systemen).
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** . 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

Fügen Sie ein CA-Zertifikat mit folgenden iKeycmd- -Befehlen hinzu:

- Geben Sie unter UNIX, Linux und Windowsden folgenden Befehl aus:

```
runmqckm -cert -add -db filename -pw password -label label -file filename
        -format ascii
```

Dabei gilt:

`-db` *`filename`*	ist der vollständig qualifizierte Pfadname der CMS-Schlüsseldatenbank.
`-pw` *`password`*	ist das Kennwort für die CMS-Schlüsseldatenbank.
`-label` *`label`*	Ist der Kennsatz, der dem Zertifikat zugeordnet ist.
`-file` *`filename`*	ist der Name der Datei, die das Zertifikat enthält.
`-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`.

Persönliche Zertifikate aus einem Schlüsselrepository exportieren

Gehen Sie wie folgt vor, um ein persönliches Zertifikat zu exportieren.

iKeyman verwenden

Wenn Sie SSL-Zertifikate FIPS-konform verwalten müssen, verwenden Sie den Befehl 'runmqakm'. 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 iKeyman -GUI mit dem Befehl **`strmqikm`** (unter Windows UNIX and Linux).
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

Mit den folgenden Befehlen können Sie unter Verwendung von iKeycmd ein persönliches Zertifikat exportieren:

- Unter UNIX, Linux und Windows:

```
runmqckm -cert -export -db filename -pw password -label label -type cms
        -target filename -target_pw password -target_type pkcs12
```

Dabei gilt:

`-db` *`filename`*	ist der vollständig qualifizierte Pfadname der CMS-Schlüsseldatenbank.
`-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.

Persönliches Zertifikat in ein Schlüsselrepository auf UNIX, Linux, and Windows -Systemen importieren

Gehen Sie wie folgt vor, um ein persönliches Zertifikat zu importieren:

Bevor Sie ein persönliches Zertifikat in das PKCS#12-Format in die Schlüsseldatenbankdatei importieren, müssen Sie zuerst die vollständige gültige Kette für die Ausgabe von CA-Zertifikaten zur Schlüsseldatenbankdatei hinzufügen (siehe „CA-Zertifikat (oder den öffentlichen Teil eines selbst signierten Zertifikats) in einem Schlüsselrepository auf UNIX, Linux, and Windows -Systemen hinzufügen” auf Seite 139).

PKCS#12-Dateien sollten als temporär betrachtet und nach der Verwendung gelöscht werden.

iKeyman verwenden

Wenn Sie SSL-Zertifikate FIPS-konform verwalten müssen, verwenden Sie den Befehl runmqakm. iKeyman 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 GUI iKeyman 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. In diesem Fenster können die Bezeichnungen der zu importierenden Zertifikate geändert werden, wenn beispielsweise bereits ein Zertifikat mit derselben Bezeichnung in der Zielschlüsseldatenbank vorhanden ist. Das Ändern der Zertifikatsbezeichnungen hat keine Auswirkungen auf die Validierung der Zertifikatskette. Dies kann verwendet werden, um die Bezeichnung des persönlichen Zertifikats in die für WebSphere MQ erforderliche Bezeichnung zu ändern, damit das Zertifikat dem bestimmten Warteschlangenmanager oder Client (z. B.ibmwebspheremqqm1) zugeordnet werden kann.
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 'Bezeichnungen ändern' wird geschlossen und das ursprüngliche Fenster ' IBM Key Management' wird erneut mit den Feldern **Persönliche Zertifikate** und **Unterzeichnerzertifikate** angezeigt, die mit den korrekt beschrifteten Zertifikaten aktualisiert wurden.
19. Das Zertifikat wird in die Zielschlüsseldatenbank importiert.

Verwenden der Befehlszeile

Mit den folgenden Befehlen können Sie unter Verwendung von iKeycmd ein persönliches Zertifikat importieren:

- Unter UNIX, Linux und Windows:

```
runmqckm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label
```

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.

iKeycmd stellt keinen Befehl zum direkten Ändern von Zertifikatskennsätzen bereit. 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 mit dem Befehl **`-cert -delete`** aus der ursprünglichen Schlüsseldatenbank.
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.

Aus einer Microsoft .pfx-Datei importieren

Führen Sie diese Prozedur aus, um eine Microsoft .pfx-Datei mit iKeymanzu portieren. Sie können runmqakm nicht verwenden, um eine PFX-Datei 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 iKeyman -GUI mit dem Befehl **strmqikm** (unter Linux, UNIX oder Windows). 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 Anzeigename muss das Format `ibmwebspheremqha`ben, gefolgt vom Namen des Warteschlangenmanagers oder der Anmelde-ID des WebSphere MQ MQI-Clientbenutzers in Kleinbuchstaben.

21. Klicken Sie auf **OK**. Das Fenster 'Change Labels' (Bezeichnungen ändern) wird jetzt entfernt und das ursprüngliche Fenster ' IBM Key Management' wird erneut mit den Anzeigen 'Personal Certificates' (Persönliche Zertifikate) und 'Signer Certificates ' (Unterzeichnerzertifikate) angezeigt, die mit dem korrekt gekennzeichneten persönlichen Zertifikat aktualisiert wurden.
22. Das persönliche PFX-Zertifikat wird nun in die (Ziel-) Datenbank importiert.

Es ist nicht möglich, eine Zertifikatsbezeichnung mit iKeycmd

Aus einer PKCS #7-Datei importieren

Die Tools iKeyman und iKeycmd unterstützen keine PKCS #7 -Dateien (.p7b). Verwenden Sie das Tool runmqckm, um Zertifikate aus einer PKCS #7-Datei zu importieren.

Verwenden Sie den folgenden Befehl, um ein CA-Zertifikat aus einer PKCS #7-Datei hinzuzufügen:

```
runmqckm -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:

```
runmqckm -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.

Zertifikat aus einem Schlüsselrepository auf UNIX, Linux, and Windows -Systemen löschen

Verwenden Sie diese Prozedur, um persönliche Zertifikate oder CA-Zertifikate zu entfernen.

iKeyman verwenden

Wenn Sie SSL-Zertifikate FIPS-konform verwalten müssen, verwenden Sie den Befehl 'runmqakm'. iKeyman stellt keine FIPS-kompatible Option bereit.

1. Starten Sie die grafische Benutzerschnittstelle iKeyman mit dem Befehl **strmqikm** (auf UNIX-, Linux -und Windows -Systemen).
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 Sie diese speichern möchten, klicken Sie auf **Exportieren/Importieren** und exportieren Sie sie (siehe „Persönliche Zertifikate aus einem Schlüsselrepository exportieren” auf Seite 140).
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

Mit den folgenden Befehlen können Sie unter Verwendung von iKeycmd oder runmqakm ein Zertifikat löschen:

- Unter UNIX, Linux und Windows:

```
runmqckm -cert -delete -db filename -pw password -label label
```

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. Dieser Modus deaktiviert die Verwendung der BSafe-Verschlüsselungsbibliothek. Im FIPS-Modus wird nur die ICC-Komponente verwendet, die für diesen Modus erfolgreich initialisiert werden muss. 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.

Sichere Kennwörter zum Schutz des Schlüsselrepositorys generieren

Sie können mit dem Befehl **runmqakm** sichere Kennwörter für den Schutz des Schlüsselrepositorys generieren.

Sie können den Befehl **runmqakm** mit den folgenden Parametern verwenden, um ein sicheres 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 UNIX and Linux -Systemen müssen Sie auch ein Backslasch-Zeichen verwenden, um die folgenden Zeichen zu entweichen, wenn sie in der Kennwortzeichenfolge vorkommen:

```
!   \   "    '
```

Wenn Sie das Kennwort als Antwort auf eine Eingabeaufforderung von **runmqckm**, **runmqakm** oder der iKeyman -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.

Verschlüsselungshardware auf UNIX, Linux, and Windows -Systemen konfigurieren

Sie können Verschlüsselungshardware für einen WS-Manager oder Client auf verschiedene Arten konfigurieren.

Sie können Verschlüsselungshardware für einen Warteschlangenmanager auf UNIX-, Linux -oder Windows -Systemen 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 WebSphere MQ Explorer, um die Verschlüsselungshardware auf Ihrem UNIX-, Linux -oder Windows -System zu konfigurieren. Weitere Informationen finden Sie in der Onlinehilfe.

Sie können Verschlüsselungshardware für einen WebSphere MQ auf UNIX-, Linux -oder Windows -Systemen mit einer der folgenden Methoden konfigurieren:

- 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_PCS11-Version des Parameters SSLCRYP verwenden, muss der Kennsatz PKCS #11 vollständig in Kleinbuchstaben angegeben werden.
- Setzen Sie das Feld **CryptoHardware** der SSL-Konfigurationsoptionsstruktur (MQSCO) in einem MQCONNX-Aufruf. Weitere Informationen finden Sie im Abschnitt Übersicht für MQSCO .

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. Weitere Informationen hierzu finden Sie im Abschnitt „Zertifikate auf PKCS #11-Hardware verwalten” auf Seite 147.

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 WebSphere 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:

   - Auf UNIX, Linux, and Windows -Systemen:

     ```
     runmqckm -keydb -create -db filename -pw password -type cms -stash
     ```

   - runmqakm wird verwendet:

     ```
     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 WebSphere MQmuss dies `cms`sein.)

   -stash
   Speichert das Kennwort der Schlüsseldatenbank in einer Datei.

   -fips
   Inaktiviert die Verwendung der BSafe-Verschlüsselungsbibliothek. Im FIPS-Modus wird nur die ICC-Komponente verwendet, die für diesen Modus erfolgreich initialisiert werden muss. 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 befinden sich im standardmäßigen druckbaren ASCII-Zeichensatz im Bereich von 0x20 bis 0x7E.

Alternativ können Sie eine Schlüsseldatenbank über die Benutzerschnittstelle von **strmqikm** (iKeyman) erstellen.

2. Melden Sie sich auf UNIX 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 von iKeyman , indem Sie den Befehl **strmqikm** ausführen.
4. Klicken Sie auf **Schlüsseldatenbankdatei** > **Öffnen**.
5. Klicken Sie auf **Schlüsseldatenbanktyp** und wählen Sie **PKCS11Direct** aus.
6. 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, ist zu beachten, dass es sich bei iKeycmd und iKeyman um 64-Bit-Programme handelt. Externe Module, die für eine PKCS#11-Unterstützung erforderlich sind, werden in einen 64-Bit-Pro-

zess geladen, daher muss für die Verwaltung der Verschlüsselungshardware eine 64-Bit-PKCS#11-Bibliothek installiert sein. Die 32-Bit-Plattformen von Windows und Linux x86 sind die einzigen Ausnahmen, da die Programme iKeyman und iKeycmd auf diesen Plattformen 32-Bit sind.

7. Geben Sie in das Feld **Position** den Pfad ein:
   - Auf UNIX 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.
8. Geben Sie im Feld **Cryptographic Token Password** das Kennwort ein, das Sie bei der Konfiguration der Verschlüsselungshardware festgelegt haben.
9. 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 „13” auf Seite 149 fort.

   Wenn Sie eine sekundäre CMS-Schlüsseldatenbank benötigen, um die Unterzeichnerzertifikate zu speichern, wählen Sie entweder **Vorhandene Sekundärschlüsseldatenbankdatei öffnen** oder **Neue Sekundärschlüsseldatenbankdatei erstellen** aus.
10. 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.
11. Geben Sie in das Feld **Position** den Pfad ein, z. B.:
    - Für einen Warteschlangenmanager: `/var/mqm/qmgrs/QM1/ssl`
    - Für einen IBM WebSphere MQ MQI-Client: `/var/mqm/ssl`

    Klicken Sie auf **OK**. Das Fenster "Password Prompt" wird geöffnet.
12. Geben Sie ein Kennwort ein.

    Wenn Sie in Schritt „9” auf Seite 149 die Option **Vorhandene Sekundärschlüsseldatenbankdatei öffnen** ausgewählt haben, geben Sie im Feld **Kennwort** ein Kennwort ein.

    Wenn Sie in Schritt „9” auf Seite 149 **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. Beachten Sie, dass wenn Sie das Kennwort nicht verdeckt speichern, alle Versuche, einen SSL-Kanal zu starten, fehlschlagen, da kein Kennwort für den Zugriff auf die Schlüsseldatenbankdatei abgerufen werden kann.

    c) Klicken Sie auf **OK**. Es wird ein Fenster geöffnet, in dem bestätigt wird, dass das Kennwort in der Datei `key.sth` enthalten ist (es sei denn, Sie haben einen anderen Stammnamen angegeben).
13. Klicken Sie auf **OK**. Der Inhaltsrahmen für die Schlüsseldatenbank wird angezeigt.

Anfordern eines persönlichen Zertifikats für Ihre PKCS #11-Hardware
Verwenden Sie diese Prozedur für einen WS-Manager oder einen IBM WebSphere MQ MQI-Client, um ein persönliches Zertifikat für Ihre Verschlüsselungshardware anzufordern.

iKeyman-Benutzerschnittstelle verwenden

Informationen zu diesem Vorgang

Anmerkung: WebSphere 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 iKeyman-Benutzerschnittstelle anzufordern:

1. Führen Sie die Schritte aus, um mit der Verschlüsselungshardware zu arbeiten. Siehe „Zertifikate auf PKCS #11-Hardware verwalten” auf Seite 147.
2. Klicken Sie im Menü **Erstellen** auf **Neue Zertifikatsanforderung** .

   Das Fenster "Neuen Schlüssel-und Zertifikatsanforderung erstellen" wird geöffnet.
3. Geben Sie im Feld **Schlüsselkennsatz** die folgenden Kennsätze ein:
   - Geben Sie für einen Warteschlangenmanager `ibmwebspheremq` gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben ein. Geben Sie für einen Warteschlangenmanager mit dem Namen `QM1`beispielsweise `ibmwebspheremqqm1`ein.
   - Für eine IBM WebSphere MQ MQI clientgeben Sie `ibmwebspheremq` gefolgt von Ihrer Anmelde-Benutzer-ID in Kleinbuchstaben ein, z. B. `ibmwebspheremqmyuserid` .
4. 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 11.
5. 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.
6. Klicken Sie auf **OK**.

   Ein Bestätigungsfenster wird geöffnet.
7. 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 Datei gespeichert, die Sie in Schritt „5” auf Seite 150 ausgewählt haben.
8. 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 -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_email-addr` *`email_addresses`* oder `-san_ipaddr` *`IP_addresses`* verwenden.

- runmqakm wird verwendet:

```
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.

-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.

-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. Dieser Modus deaktiviert die Verwendung der BSafe-Verschlüsselungsbibliothek. Im FIPS-Modus wird nur die ICC-Komponente verwendet, die für diesen Modus erfolgreich initialisiert werden muss. 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. Mögliche Werte sind 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, SHAWithDSAoder 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. Mögliche Werte: 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 .

-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.

Persönlichem Zertifikat in Ihre PKCS #11-Hardware importieren
Verwenden Sie diese Prozedur für einen Warteschlangenmanager oder einen IBM WebSphere MQ MQI-Client, um ein persönliches Zertifikat in Ihre Verschlüsselungshardware zu importieren.

iKeyman verwenden

Vorgehensweise

Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat von der iKeyman-Benutzerschnittstelle anzufordern:

1. Führen Sie die Schritte aus, um mit der Verschlüsselungshardware zu arbeiten. Siehe „Zertifikate auf PKCS #11-Hardware verwalten” auf Seite 147.
2. Klicken Sie auf **Empfangen** . Das Fenster 'Receive Certificate from a File' (Zertifikat aus einer Datei empfangen) wird angezeigt.
3. Wählen Sie den **Datentyp** des neuen persönlichen Zertifikats aus, z. B. `Base64-encoded ASCII data` für eine Datei mit der Erweiterung " `.arm` ".
4. 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.
5. 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.
6. Klicken Sie auf **Ja** oder **Nein**. Das Fenster "Enter a Label" wird geöffnet.
7. Geben Sie eine Bezeichnung ein.

   Sie können beispielsweise dieselbe Bezeichnung wie bei der Anforderung des persönlichen Zertifikats verwenden. Beachten Sie, dass die Bezeichnung das richtige IBM WebSphere MQ -Format haben muss:

   - Für einen Warteschlangenmanager `ibmwebspheremq` , gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben. Für einen Warteschlangenmanager mit dem Namen QM1lautet die Bezeichnung beispielsweise `ibmwebspheremqqm1`.
   - Bei einem IBM WebSphere MQ MQI-Client `ibmwebspheremq` gefolgt von Ihrer Anmelde-Benutzer-ID in Kleinbuchstaben. Beispiel: Für die Benutzer-ID `MyUserID`lautet die Bezeichnung `ibmwebspheremqmyuserid`.
8. 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.

Verwenden der Befehlszeile

Vorgehensweise

Führen Sie die folgenden Schritte aus, um ein persönliches Zertifikat über eine Befehlszeile anzufordern:

1. Öffnen Sie ein Befehlsfenster, das für Ihre Umgebung konfiguriert ist.
2. Geben Sie den entsprechenden Befehl für Ihr Betriebssystem und Ihre Konfiguration ein:

   - Verwenden Sie auf Windows-, UNIX and Linux -Systemen einen der folgenden Befehle:

     ```
     runmqckm -cert -receive -file filename -crypto path
     -tokenlabel hardware_token -pw hardware_password -format cert_format
     ```

     ```
     runmqakm -cert -receive -file filename -crypto path
     -tokenlabel hardware_token -pw hardware_password -format cert_format -fips
     ```

   Dabei gilt:

   -file *Dateiname*
   Gibt den vollständig qualifizierten Dateinamen der Datei an, die das persönliche Zertifikat enthält.

   -crypto *Pfad*
   Gibt den vollständig qualifizierten Pfad zur PKCS #11-Bibliothek an, die mit der Hardware geliefert wird.

-tokenlabel *Hardware-Token*
Gibt die Bezeichnung an, die dem Speicherteil der Verschlüsselungshardware während der Installation gegeben wird.

-pw *Hardware-Kennwort*
Gibt das Kennwort für den Zugriff auf die Hardware 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. Dieser Modus inaktiviert die Verwendung der BSafe-Verschlüsselungsbibliothek. Im FIPS-Modus wird nur die ICC-Komponente verwendet, die für diesen Modus erfolgreich initialisiert werden muss. 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.

Benutzer identifizieren und authentifizieren

Sie können Benutzer mithilfe der MQCSP-Struktur oder in mehreren Typen von Benutzerexitprogrammen identifizieren und authentifizieren.

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.

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 agiert ein MCA normalerweise im Namen des Benutzers der WebSphere MQ -Clientanwendung. In dieser Situation erfolgt die gegenseitige Authentifizierung zwischen zwei Warteschlangenmanagern oder zwischen einem Warteschlangenmanager und dem Benutzer einer WebSphere MQ MQI-Clientanwendung.

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 „Das SSPI-Kanalexitprogramm” auf Seite 109.

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.

Secure Sockets Layer (SSL) und Transport Layer Security (TLS) verwenden ähnliche PKI-Verfahren wie hier beschrieben. Weitere Informationen dazu, wie SSL und TLS eine Authentifizierung durchführen, finden Sie in „Secure Sockets Layer (SSL) and Transport Layer Security (TLS) concepts" auf Seite 15.

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 238 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 79 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 im Abschnitt „Identitätsabgleich im API-Exit und API-Steuerübergabeexit" auf Seite 158.

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.

Privilegierte Benutzer

Ein privilegierter Benutzer ist ein Benutzer mit vollständigen Administratorberechtigungen für WebSphere MQ.

Neben den in der folgenden Tabelle aufgelisteten Benutzern sind Mitglieder jeder Gruppe mit `+crt` -Berechtigung für Warteschlangen indirekt Administratoren. Ebenso ist jeder Benutzer, der über die Berechtigung `+set` für den Warteschlangenmanager und die Berechtigung `+put` für die Befehlswarteschlange verfügt, ein Administrator.

Diese Berechtigungen sollten Sie Standardbenutzern und -anwendungen nicht erteilen.

Tabelle 13. Privilegierte Benutzer nach Plattform.

Eine Tabelle mit privilegierten Benutzern. Unter Windowssind SYSTEM alle Mitglieder der Gruppe mqm und alle Mitglieder der Gruppe Administratoren privilegierte Benutzer. Auf UNIX and Linux-Systemen sind alle Mitglieder der Gruppe 'mqm' als privilegierte Benutzer definiert. Unter IBM i sind die Profile (Benutzer) qmqm und qmqmadm, alle Mitglieder der Gruppe qmqmadm und alle Benutzer, die mit der Einstellung *ALLOBJ definiert sind, privilegierte Benutzer.

Plattform	Privilegierte Benutzer
Windows-Systeme	• SYSTEM • Mitglieder der Gruppe 'mqm' • Mitglieder der Gruppe Administratoren
UNIX and Linux-Systeme	• Mitglieder der Gruppe 'mqm'

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.

Die mit IBM WebSphere MQ bereitgestellte Berechtigungsservicekomponente wird als OAM (Object Authority Manager, Objektberechtigungsmanager) bezeichnet. Der OAM berechtigt Benutzer auf der Basis der ID, die im MQCSP enthalten ist, aber das Kennwort nicht. Es ist möglich, die Kennwortprüfung im Authorization Service durch die Verwendung von verketteten Exits mit dem OAM zu implementieren, oder indem der OAM durch einen alternativen Berechtigungsservice ersetzt wird.

Sie können den MQCSP in einem Sicherheitsexit ändern.

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 agiert ein MCA normalerweise im Namen des Benutzers der WebSphere MQ MQI-Clientanwendung. In dieser Situation erfolgt die gegenseitige Authentifizierung

zwischen zwei Warteschlangenmanagern oder zwischen einem Warteschlangenmanager und dem Benutzer einer WebSphere MQ MQI-Clientanwendung.

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 in „Das SSPI-Kanalexitprogramm" auf Seite 109.

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.

Secure Sockets Layer (SSL) und Transport Layer Security (TLS) verwenden ähnliche PKI-Verfahren wie hier beschrieben. Weitere Informationen zur Authentifizierung von Secure Sockets Layer finden Sie in „Secure Sockets Layer (SSL) and Transport Layer Security (TLS) concepts" auf Seite 15.

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 238 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 79 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 158.

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 SSL-Handshake authentifizieren sich die beiden Kommunikationspartner gegenseitig mithilfe digitaler Zertifikate. 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.

Auf UNIX-, Linux -und Windows -Systemen überprüft die SSL-Unterstützung von WebSphere MQ mithilfe von OCSP (Online Certificate Status Protocol) oder mithilfe von CRLs und ARLs auf LDAP-Servern (Lightweight Directory Access Protocol) auf widerrufene Zertifikate. OCSP ist die bevorzugte Methode. IBM WebSphere MQ classes for Java und IBM WebSphere MQ classes for JMS können die OCSP-Informationen in einer Clientkanaldefinitionstabellendatei nicht verwenden. Allerdings kann OCSP, wie im Abschnitt Online Certificate Protocol verwenden beschrieben, konfiguriert werden.

Unter z/OS und IBM i unterstützt WebSphere MQ SSL-Unterstützungsprüfungen für widerrufene Zertifikate nur unter Verwendung von CRLs und ARLs auf LDAP-Servern.

Weitere Informationen zum Zertifikat

Berechtigungen, siehe „Digitale Zertifikate” auf Seite 10.

Widerruftes Zertifikat und OCSP

IBM WebSphere 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 WebSphere MQ auf Windows- und UNIX and Linux-Systemen.

Um den Widerrufsstatus eines digitalen Zertifikats mit OCSP zu überprüfen, bestimmt WebSphere MQ mit zwei Methoden, 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- oder SSL-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 WebSphere MQ kann nicht auf den OCSP-Responder zugreifen.
- Der OCSP-Responder hat eine Antwort gesendet, WebSphere 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 WebSphere MQ das OCSP-Ergebnis `Unbekannt` empfängt, hängt sein Verhalten von der Einstellung des Attributs 'OCSPAuthentication' ab. Bei Warteschlangenmanagern befindet sich dieses Attribut in der SSL-Zeilengruppe der Datei `qm.ini` für UNIX and Linux -Systeme oder in der Windows -Registry. Sie kann mit IBM WebSphere MQ Explorer festgelegt werden. Für Clients ist es in der SSL-Zeilengruppe der Clientkonfigurationsdatei enthalten.

Wenn das Ergebnis `Unbekannt` empfangen wird und 'OCSPAuthentication' auf REQUIRED gesetzt ist (der Standardwert), lehnt WebSphere 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, lässt WebSphere MQ das Starten des SSL-Kanals zu 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 WebSphere 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 konfigurieren. Die Schritte, die Sie zum Herstellen der SSL-Verbindung ausgeführt haben, sind für die Überprüfung der OCSP-Antwort ausreichend.
- 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 Zertifikat gesendet wird, mit dem sie unterzeichnet wurde (und dieses Zertifikat von einer Zertifizierungsstelle unterzeichnet wurde, die bereits für die SSL-Verbindung 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. Sie müssen eine Kopie des OCSP-Responder-Zertifikats zur Schlüsseldatenbank des Clients oder Warteschlangenmanagers hinzufügen, der die OCSP-Prüfung ausführt. Weitere Informationen finden Sie unter „CA-Zertifikat (oder den öffentlichen Teil eines selbst signierten Zertifikats) in einem Schlüsselrepository auf UNIX, Linux, and Windows -Systemen hinzufügen” auf Seite 139. Wenn ein CA-Zertifikat hinzugefügt wird, wird es standardmäßig als Trusted Root hinzugefügt. Dies ist die erforderliche Einstellung in diesem Kontext. Wenn dieses Zertifikat nicht hinzugefügt wird, kann WebSphere MQ die digitale Signatur der OCSP-Antwort und die OCSP-Prüfungsergebnisse in einem Ergebnis vom Typ 'Unbekannt' nicht überprüfen, was dazu führen kann, dass IBM WebSphere MQ je nach OCSPAuthentication-Wert den Kanal schließt.

Online Certificate Status Protocol (OCSP) in Java- und JMS-Clientanwendungen

Aufgrund von Einschränkungen der Java-API kann WebSphere MQ die Überprüfung des OCSP-Zertifikatswiderrufs (Online Certificate Status Protocol) für sichere SSL- und TLS-Sockets nur dann 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 java.security.Security.setProperty()-API entsprechend der jeweils gültigen Java Security Manager-Richtlinie.

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.

Eigenschaftsname	Beschreibung
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.
ocsp.responderCertSubjectName	Der Wert dieses Merkmals ist der Betreffname des Zertifikats des OCSP-Responderers. 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 manchen Fällen kann diese Konfiguration unerwünschte Nebenwirkungen zeigen, wenn die JVM auch von einem anderen Anwendungscode verwendet wird, der sichere SSL- oder TLS-Sockets verwendet. 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 daher bei der Anwendung vorläufiger Java-Fixes und Produktwartungen 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 unter OCSP-Prüfung in nativen Abfangprozessen aktivieren beschriebenen AMS-Java-Abfangprozess verwenden, sollten Sie keine 'java.security'-OCSP-Konfiguration verwenden, die mit der AMS-OCSP-Konfiguration in der Schlüsselspeicherkonfigurationsdatei in Konflikt steht.

Mit Zertifikatswiedergabelisten und Berechtigungslisten für die Berechtigung arbeiten

Die Unterstützung von WebSphere MQfür CRLs und ARLs variiert je nach Plattform.

Die CRL-und ARL-Unterstützung auf jeder Plattform ist wie folgt:

- Unter z/OSunterstützt System SSL CRLs und ARLs, die von Tivoli Public Key Infrastructure auf LDAP-Servern gespeichert werden.
- Auf anderen Plattformen entspricht die CRL- und ARL-Unterstützung den Empfehlungen für PKIX-CRL-Profile gemäß Standard X.509 V2.

WebSphere MQ verwaltet einen Cache mit CRLs und ARLs, auf die in den letzten 12 Stunden zugegriffen wurde.

Wenn ein Warteschlangenmanager oder WebSphere MQ MQI-Client ein Zertifikat empfängt, prüft er die CRL, um sicherzustellen, dass das Zertifikat noch gültig ist. WebSphere MQ prüft zuerst im Cache, ob ein Cache vorhanden ist. Wenn sich die CRL nicht im Cache befindet, fragt WebSphere MQ die LDAP-CRL-Serverpositionen in der Reihenfolge ab, in der sie in der Namensliste der Authentifizierungsinformationsobjekte auftreten, die mit dem Attribut *SSLCRLNamelist* angegeben sind, bis WebSphere MQ eine verfügbare CRL findet. Wenn die Namensliste nicht angegeben ist oder mit einem Leerwert angegeben wird, werden CRLs nicht überprüft.

Weitere Informationen zu LDAP finden Sie unter Using lightweight directory access protocol services with WebSphere MQ for Windows.

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. Zur Verwendung mit WebSphere MQ SSL müssen die Binärdaten für diese Attribute dem Format DER (Definite Encoding Rules) entsprechen. Weitere Informationen zu LDIF-Dateien finden Sie in der Dokumentation, die mit dem LDAP-Server bereitgestellt wird.

Abbildung 12 auf Seite 164 zeigt eine LDIF-Beispieldatei, die Sie als Eingabe für Ihren LDAP-Server erstellen können, um die von CA1ausgegebenen CRLs und ARLs zu laden. Hierbei handelt es sich um eine imaginäre Zertifizierungsstelle mit dem definierten Namen "CN=CA1, OU=Test, O=IBM, C=GB", die von der Testorganisation in IBMeingerichtet 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 12. LDIF-Beispieldatei für eine Zertifizierungsstelle. Dies kann von der Implementierung bis zur Implementierung variieren.

Abbildung 13 auf Seite 164 zeigt die DIT-Struktur, die Ihr LDAP-Server erstellt, wenn Sie die in Abbildung 12 auf Seite 164 gezeigte LDIF-Beispieldatei zusammen mit einer ähnlichen Datei für CA2, eine imaginäre Zertifizierungsstelle, die von der PKI-Organisation eingerichtet wurde, ebenfalls in IBMladen.

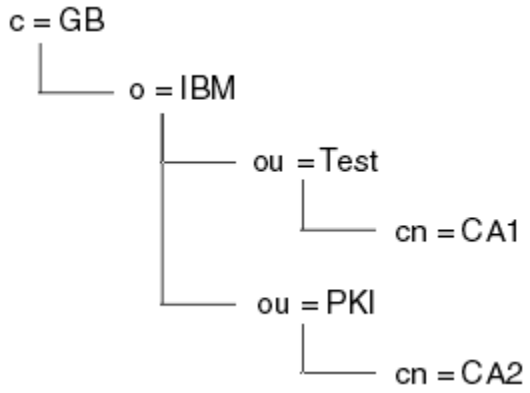


Abbildung 13. Beispiel für eine Struktur des LDAP-Verzeichnisinformationsbaums

WebSphere 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. WebSphere 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 164 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.

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 gehalten, die im WS-Manager-Attribut *SSLCRLNamelist* 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.

   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.

   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.
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 WS-Manager-Attribut mit dem Namen *SSLCRLNamelist* festgelegt. Der Anfangswert des WS-Managers für dieses Attribut ist leer.

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 *müssen* identische Informationen enthalten.

Mit IBM WebSphere MQ Explorer auf CRLs und ARLs zugreifen

Sie können IBM WebSphere MQ Explorer verwenden, um einen Warteschlangenmanager zu informieren, wie auf CRLs zugegriffen werden kann.

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 unter **Eigenschaften ändern** auf der Seite **Allgemein** 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 dann auf **Neu-> Namensliste** . In dem Eigenschaftenblatt, das geöffnet wird:

a. Geben Sie einen Namen für die Namensliste ein.

b. Fügen Sie den Namen des CRL-Objekts (LDAP) (aus Schritt „2.a" auf Seite 165) in die 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 den Namen der Namensliste (aus Schritt „3.a" auf Seite 166) in das Feld **CRL-Namensliste** ein.

Mit einem IBM WebSphere MQ MQI-Client auf CRLs und ARLs zugreifen

Sie haben drei Optionen zur Angabe der LDAP-Server, die CRLs für die Überprüfung durch einen IBM WebSphere 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
- Active Directory verwenden (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 *müssen* identische Informationen enthalten.

Sie können über einen WebSphere MQ MQI-Clientkanal, der auf Linux (zSeries -Plattform) ausgeführt wird, nicht auf LDAP-CRLs zugreifen.

Position eines OCSP-Responders und von LDAP-Servern, die CRLs enthalten

Auf einem IBM WebSphere MQ MQI-Clientsystem können Sie die Position eines OCSP-Responders sowie von LDAP-Servern (Lightweight Directory Access Protocol) angeben, die Zertifikatswiderrufslisten (CRLs) enthalten.

Sie können diese Positionen auf drei Arten angeben, die hier in der Reihenfolge der Rangfolge aufgelistet werden.

Wenn eine WebSphere MQ MQI-Clientanwendung 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 WebSphere MQ MQI-Client benötigt, um auf einen OCSP-Responder oder einen

LDAP-Server mit CRLs zuzugreifen. 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 WebSphere MQ MQI-Client auf einen OCSP-Responder oder LDAP-Server zugreifen kann, die CRLs enthalten, schließen Sie die Attribute eines oder mehrerer Authentifizierungsinformationsobjekte in eine Definitionstabelle für Clientkanäle 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.

Ein Authentifizierungsdatenobjekt mit AUTHTYPE (OCSP) gilt nicht für die Verwendung auf IBM i -oder z/OS -Warteschlangenmanagern, aber es kann auf diesen Plattformen angegeben werden, um in die Definitionstabelle für Clientkanäle (CCDT) für die Clientverwendung kopiert zu werden.

Um einem WebSphere MQ MQI-Client den Zugriff auf einen OCSP-Responder oder LDAP-Server zu ermöglichen, die CRLs enthalten, können die Attribute eines oder mehrerer Authentifizierungsinformationsobjekte in eine Definitionstabelle für Clientkanäle eingeschlossen werden. Sie können solche Attribute auf eine der folgenden Arten einschließen:

Auf den Serverplattformen AIX, HP-UX, Linux, Solaris und Windows

Sie können eine Namensliste definieren, die die Namen von einem oder mehreren Authentifizierungsinformationsobjekten enthält. Anschließend können Sie das WS-Manager-Attribut **SSLCRLNameList** 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. Ist ein LDAP-Server nicht verfügbar, wenn er erforderlich ist, kann ein WebSphere MQ MQI-Client versuchen, auf einen anderen zuzugreifen.

Die Attribute der Authentifizierungsinformationsobjekte, die von der Namensliste identifiziert werden, werden hier zusammen als *Zertifikatswiderrufsposition* bezeichnet. Wenn Sie das WS-Managerattribut **SSLCRLNameList** auf den Namen der Namensliste setzen, wird die Position des Zertifikatwiderrufs in die Definitionstabelle für den Clientkanal kopiert, die dem Warteschlangenmanager zugeordnet ist. Wenn auf die CCDT von einem Clientsystem aus als gemeinsam genutzte Datei zugegriffen werden kann oder wenn die CCDT anschließend auf ein Clientsystem kopiert wird, kann der WebSphere MQ MQ MQI-Client auf diesem System die Zertifikatswiderrufsposition in der CCDT verwenden, um auf einen OCSP-Responder oder auf LDAP-Server zuzugreifen, die CRLs enthalten.

Wenn die Zertifikatswiderrufposition des WS-Managers später geändert wird, wird die Änderung in der CCDT wiedergegeben, die dem Warteschlangenmanager zugeordnet ist. Wenn das WS-Managerattribut **SSLCRLNameList** auf "Leer" gesetzt ist, wird die Position des Zertifikatwiderrufs 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.

Active Directory unter Windows verwenden

Auf Windows -Systemen können Sie mit dem Steuerbefehl **setmqcrl** die aktuellen CRL-Informationen in Active Directoryveröffentlichen.

Befehl **setmqcrl** veröffentlicht keine OCSP-Informationen.

Informationen zu diesem Befehl und seiner Syntax finden Sie in setmqcrl .

Mit IBM WebSphere MQ -Klassen für Java und IBM WebSphere MQ -Klassen für JMS auf CRLs und ARLs zugreifen

IBM WebSphere MQ Classes for Java-und IBM WebSphere MQ Classes for JMS-Zugriffs-CRLs unterscheiden sich von anderen Plattformen.

Informationen zum Arbeiten mit CRLs und ARLs mit IBM WebSphere MQ -Klassen für Java finden Sie unter Zertifikatswiderrufslisten verwenden.

Informationen zum Arbeiten mit CRLs und ARLs mit IBM WebSphere MQ Classes for JMS finden Sie unter Objekteigenschaft SSLCERTSTORES.

Authentifizierungsinformationsobjekte bearbeiten

Sie können Authentifizierungsinformationsobjekte mit Hilfe von MQSC-oder PCF-Befehlen oder mit dem IBM WebSphere 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 unter Scriptbefehle (MQSC) .

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 unter Definitionen der Programmable Command Formats .

Auf Plattformen, auf denen es verfügbar ist, können Sie auch den WebSphere MQ Explorer verwenden.

Autorisieren des Zugriffs auf Objekte

Dieser Abschnitt enthält Informationen zur Verwendung des Objektberechtigungsmanagers und des Kanalexitprogramms, um den Zugriff auf Objekte zu steuern.

Auf UNIX, Linux, and Windows -Systemen. Sie können den Zugriff auf Objekte steuern, indem Sie den Objektberechtigungsmanager (OAM) verwenden. Diese Themensammlung enthält Informationen zur Verwendung der Befehlsschnittstelle für den OAM. Sie enthält außerdem eine Prüfliste, mit deren Hilfe Sie feststellen können, welche Tasks ausgeführt werden müssen, um die Sicherheit auf Ihrem System anzuwenden, und führt die Aspekte auf, die zu berücksichtigen sind, wenn Sie Benutzern Berechtigungen zum Verwalten von IBM WebSphere MQ und Arbeiten mit IBM WebSphere MQ-Objekten erteilen. Wenn

die bereitgestellten Sicherheitsmechanismen Ihre Anforderungen nicht erfüllen, können Sie eigene Kanalexitprogramme entwickeln.

Zugriff auf Objekte über OAM auf UNIX-, Linux -und Windows -Systemen steuern

Der Objektberechtigungsmanager (OAM) stellt eine Befehlsschnittstelle zum Erteilen und Entziehen von Berechtigungen für WebSphere MQ -Objekte bereit.

Zur Verwendung dieser Befehle benötigen Sie die entsprechenden Berechtigungen (siehe „Berechtigung zur Verwaltung von IBM WebSphere MQ auf UNIX, Linux, and Windows -Systemen" auf Seite 208). Benutzer-IDs, die zur Verwaltung von WebSphere MQ berechtigt sind, verfügen über die *Superuserberechtigung* für den Warteschlangenmanager, d. h., Sie müssen ihnen keine weitere Berechtigung zum Absetzen von MQI-Anforderungen oder -Befehlen erteilen.

Zugriff auf ein IBM WebSphere MQ -Objekt auf UNIX, Linux, and Windows -Systemen erteilen

Mit dem Steuerbefehl **setmqaut** oder dem PCF-Befehl **MQCMD_SET_AUTH_REC** können Sie Benutzern und Benutzergruppen Zugriff auf IBM WebSphere MQ -Objekte erteilen.

Eine vollständige Definition des **setmqaut** -Steuerbefehls und seiner Syntax finden Sie unter setmqaut. 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 auf Systemen mit UNIX, Linux, and Windows verwenden" auf Seite 170.
- 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 die Berechtigung zur Verwendung eines Objekts mit dem Befehl `setmqaut` erteilt und widerrufen wird:

```
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 UNIX and Linux -Systemen widerruft dieser Befehl auch die Berechtigung "put" für alle Principals in derselben Primärgruppe wie der Benutzer "fvuser".

```
setmqaut -m saturn.queue.manager -t queue -n MyQueue -p fvuser
         -g groupa -g groupb -put
```

Befehl 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.

Generische OAM-Profile auf Systemen mit UNIX, Linux, and Windows verwenden

Mit generischen OAM-Profilen können Sie die Berechtigung eines Benutzers für viele Objekte gleichzeitig festlegen, anstatt separate **setmqaut** -Befehle für jedes einzelne Objekt absetzen zu müssen, wenn es erstellt wird.

Mit generischen Profilen im Befehl **setmqaut** können Sie eine generische Berechtigung für alle Objekte festlegen, die zu diesem Profil passen.

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 also `ABC.?EF` angeben, gilt die Berechtigung, die Sie diesem Profil erteilen, für Objekte mit dem Namen `ABC.DEF`, `ABC.CEF`, `ABC.BEF` usw.

Folgende Platzhalterzeichen stehen zur Verfügung:

?

Verwenden Sie das Fragezeichen (?) anstelle eines beliebigen einzelnen Zeichens. So gilt beispielsweise `AB.?D` 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`.

  Beispiel: `ABC.*.JKL` gilt für die Objekte `ABC.DEF.JKL`und `ABC.GHI.JKL`. (Beachten Sie, dass es **nicht** für `ABC.JKL` gilt; * gibt in diesem Kontext immer ein Qualifikationsmerkmal an.)
- Ein Zeichen in einem Qualifikationsmerkmal in einem Profilnamen, das null oder mehr Zeichen innerhalb des Qualifikationsmerkmals in einem Objektnamen entspricht.

  `ABC.DE*.JKL` gilt beispielsweise 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 z. B. `-t prcs` 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. Beispiel: `**.ABC` gibt alle Objekte mit dem endgültigen Qualifikationsmerkmal ABC an.

Anmerkung: Wenn Sie Platzhalterzeichen auf UNIX and Linux -Systemen verwenden, **müssen** Sie den Profilnamen in einfache Anführungszeichen einschließen.

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. Im Beispiel oben besitzt also die Warteschlange AB.CD die Berechtigung zum **Abrufen** (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 PCF-Befehls **MQCMD_INQUIRE_AUTH_RECS** und seiner Syntax im Abschnitt Berechtigungsdatensätze abfragen.

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: Auch wenn UNIX and Linux-Benutzer die Option `-p` bei dem Befehl **dmpmqaut** verwenden können, müssen sie beim Definieren von Berechtigungen 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 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 für WebSphere 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
```

Platzhalterzeichen in OAM-Profilen 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; * gibt in diesem Kontext immer ein Qualifikationsmerkmal an.)

- 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 z. B. `-t prcs` 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.

Anmerkung: Wenn Sie Platzhalterzeichen auf UNIX and Linux -Systemen verwenden, **müssen** Sie den Profilnamen in einfache Anführungszeichen einschließen.

Profilprioritäten

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. Im Beispiel oben besitzt also die Warteschlange AB.CD die Berechtigung zum **Abrufen** (AB.C* ist spezifischer als AB.*).

Wenn Sie generische Zeichen vergleichen, lautet die Reihenfolge der *Spezifität* :

1. ?
2. *
3. **

Speicherauszugsprofileinstellungen

Mit dem Steuerbefehl **dmpmqaut** oder dem PCF-Befehl **MQCMD_INQUIRE_AUTH_RECS** können Sie einen Speicherauszug der aktuellen Berechtigungen erstellen, die einem angegebenen Profil zugeordnet sind.

Eine vollständige Definition des **dmpmqaut** -Steuerbefehls und seiner Syntax finden Sie unter dmpmqaut. Eine vollständige Definition des **MQCMD_INQUIRE_AUTH_RECS** -PCF-Befehls und seiner 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: UNIX and Linux -Benutzer können die Option `-p` nicht verwenden. Sie 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 für WebSphere 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
```

Zugriffseinstellungen anzeigen

Mit dem Steuerbefehl **dspmqaut** oder dem PCF-Befehl **MQCMD_INQUIRE_ENTITY_AUTH** können Sie die Berechtigungen anzeigen, die ein bestimmter Principal oder eine Gruppe für ein bestimmtes Objekt hat.

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 **dmpmqaut** -Steuerbefehls und seiner Syntax finden Sie unter dmpmqaut. Eine vollständige Definition des **MQCMD_INQUIRE_ENTITY_AUTH** -PCF-Befehls und seiner Syntax finden Sie unter Inquire Entity Authority.

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
```

Zugriff auf ein IBM WebSphere MQ-Objekt ändern und entziehen

Verwenden Sie den Befehl **setmqaut** , um die Zugriffsebene zu ändern, die ein Benutzer oder eine Gruppe auf ein Objekt hat. Um den Zugriff eines bestimmten Benutzers, der Mitglied einer Gruppe mit Berechtigung ist, zu widerrufen, entfernen Sie den Benutzer aus der Gruppe.

Der Prozess, mit dem der Benutzer aus einer Gruppe entfernt wird, wird in beschrieben:

- „Gruppen unter Windows erstellen und verwalten” auf Seite 87
- „Gruppen unter HP-UX erstellen und verwalten” auf Seite 90
- „Gruppen unter AIX erstellen und verwalten” auf Seite 91
- „Gruppen unter Solaris erstellen und verwalten” auf Seite 92
- „Gruppen unter Linux erstellen und verwalten” auf Seite 93

.

Die Benutzer-ID, mit der ein IBM WebSphere MQ-Objekt erstellt wird, erhält uneingeschränkte Zugriffsberechtigungen für dieses Objekt. Wenn Sie diese Benutzer-ID aus der lokalen Gruppe 'mqm' (oder der Administratorgruppe auf Windows -Systemen) entfernen, werden diese Berechtigungen nicht entzogen. 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 Steuerbefehls setmqaut und seiner Syntax finden Sie unter setmqaut. Eine vollständige Definition des **MQCMD_INQUIRE_ENTITY_AUTH** -PCF-Befehls und seiner Syntax finden Sie unter Inquire Entity Authority.

Löschen Sie unter Windowsdie OAM-Einträge, die einem bestimmten Windows -Benutzerkonto entsprechen, bevor Sie das Benutzerprofil löschen. Die OAM-Einträge können nach dem Entfernen des Benutzerkontos nicht mehr entfernt werden.

Sicherheitszugriffsprüfungen auf Systemen mit UNIX, Linux, and Windows verhindern

Wenn keine Sicherheitsprüfungen durchgeführt werden sollen, können Sie den OAM inaktivieren. Dies kann für eine Testumgebung geeignet sein. 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:

- Stellen Sie die Umgebungsvariable MQSNOAUT des Betriebssystems vor der Erstellung eines Warteschlangenmanagers ein (wenn Sie auf diese Weise vorgehen, können Sie später keinen OAM mehr hinzufügen):

  Weitere Informationen zu den Auswirkungen der Einstellung der Variablen MQSNOAUT finden Sie unter Umgebungsvariablen .
- Bearbeiten Sie die Konfigurationsdatei des Warteschlangenmanagers, um den Service zu entfernen. (Wenn Sie dies tun, können Sie später keinen OAM hinzufügen.)

Wenn Sie den Befehl 'setmqaut' oder 'dspmqaut' ausführen, so lange der OAM inaktiviert ist, ist Folgendes zu berücksichtigen:

- Der OAM validiert den angegebenen Principal bzw. die angegebene Gruppe nicht, was 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.

Warnung: Wenn ein OAM entfernt wird, kann er nicht auf einen vorhandenen Warteschlangenmanager zurückgestellt werden. Dies liegt daran, dass der OAM bei der Objekterstellung vorhanden sein muss. Damit der OAM WebSphere MQ nach dem Entfernen wieder verwendet werden kann, muss der Warteschlangenmanager erneut erstellt werden.

Zugehörige Konzepte

Installierbare Services

Erforderlicher Zugriff auf Ressourcen erteilen

Verwenden Sie diesen Artikel, um zu bestimmen, welche Tasks ausgeführt werden müssen, um die Sicherheit auf Ihr WebSphere MQ -System anzuwenden.

Informationen zu diesem Vorgang

Während dieser Task entscheiden Sie, welche Aktionen erforderlich sind, um die entsprechende Sicherheitsstufe auf die Elemente Ihrer WebSphere 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 178.
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 184.
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 189.
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 190.
6. Benötigen Sie Benutzeranwendungen, um eine Verbindung zu Ihrem Warteschlangenmanager herzustellen?
   a) Nein: Inaktivierbare Konnektivität, wie in „Verbindung zum WS-Manager wird entfernt" auf Seite 191 beschrieben
   b) Ja: Siehe „Benutzeranwendungen die Verbindung zum Warteschlangenmanager ermöglichen" auf Seite 191.

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 14. 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 in „beschränkten Verwaltungszugriff auf einige Warteschlangen erteilen" auf Seite 179 beschrieben.
Themen	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Themen, wie in „Erteilen eines eingeschränkten Verwaltungszugriffs auf bestimmte Themen" auf Seite 180 beschrieben.

Tabelle 14. Teilweiser Verwaltungszugriff auf eine Untergruppe von WS-Manager-Ressourcen erteilen (Forts.)

Die Benutzer müssen Objekte dieses Typs verwalten.	**Diese Aktion ausführen**
Kanäle	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Kanäle, wie in „beschränkten Verwaltungszugriff auf einige Kanäle erteilen" auf Seite 180 beschrieben.
Der Warteschlangenmanager	Erteilen eines partiellen Verwaltungszugriffs auf den Warteschlangenmanager, wie in „Erteilen des eingeschränkten Verwaltungszugriffs auf einen Warteschlangenmanager" auf Seite 181 beschrieben
Prozesse	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Prozesse, wie in „Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Prozesse" auf Seite 182 beschrieben.
Namenslisten	Erteilen Sie den teilweisen Verwaltungszugriff auf die erforderlichen Namenslisten, wie in „Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Namenslisten" auf Seite 182 beschrieben.
Services	Erteilen Sie partiellen Verwaltungszugriff auf die erforderlichen Services, wie in „Erteilen eines eingeschränkten Verwaltungszugriffs auf einige Services" auf Seite 183 beschrieben.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
```

- Die Variablennamen haben die folgenden Bedeutungen:

  QMgrName
  Der Name des Warteschlangenmanagers. Unter z/OSkann 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:

– Auf UNIX-, Linux -und Windows -Systemen eine beliebige Kombination der folgenden Berechtigungen: + chg, + clr, + dlt, + dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
```

- 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:

  – Auf UNIX-, Linux -und Windows -Systemen eine beliebige Kombination der folgenden Berechtigungen: + chg, + clr, + crt, + dlt, + dsp. + strg. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- Die Variablennamen haben die folgenden Bedeutungen:

QMgrName
Der Name des Warteschlangenmanagers. Unter z/OSkann 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:

– Auf UNIX-, Linux -und Windows -Systemen eine beliebige Kombination der folgenden Berechtigungen: + chg, + clr, + crt, + dlt, + dsp. + strg, + strgx. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('QMgrNa
me')
```

Ergebnisse

Geben Sie für jeden MQSC-Befehl folgende Befehle aus, um festzulegen, welche MQSC-Befehle der Benutzer auf dem Warteschlangenmanager ausführen kann.

```
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:

- Auf UNIX-, Linux -und Windows -Systemen eine beliebige Kombination der folgenden Berechtigungen: + 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
```

- 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:

  – Auf UNIX-, Linux -und Windows -Systemen eine beliebige Kombination der folgenden Berechtigungen: + chg, + clr, + crt, + dlt, + dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- 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:

– Auf UNIX-, Linux -und Windows -Systemen eine beliebige Kombination der folgenden Berechtigungen: + chg, + clr, + crt, + dlt, + ctrl, + ctrlx, + dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

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.

Anmerkung: Unter z/OS sind keine Serviceobjekte vorhanden.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('QMgrNa
me')
```

Ergebnisse

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:

- Auf UNIX-, Linux -und Windows -Systemen eine beliebige Kombination der folgenden Berechtigungen: + chg, + clr, + crt, + dlt, + ctrl, + ctrlx, + dsp. Die Berechtigung + alladm ist äquivalent zu + chg + clr + dlt + dsp.

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 15. 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 in „Vollzugriff Verwaltungszugriff auf einige Warteschlangen erteilen” auf Seite 184 beschrieben.
Themen	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Themen, wie in „Vollenden Verwaltungszugriff auf einige Themen erteilen” auf Seite 185 beschrieben.
Kanäle	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Kanäle, wie in „Vollenden Verwaltungszugriff auf einige Kanäle erteilen” auf Seite 186 beschrieben.
Der Warteschlangenmanager	Erteilen des vollständigen Verwaltungszugriffs auf den Warteschlangenmanager, wie in „Vollzugriff auf den Verwaltungszugriff auf einen Warteschlangenmanager erteilen” auf Seite 186 beschrieben
Prozesse	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Prozesse, wie in „Vollenden Verwaltungszugriff auf einige Prozesse erteilen” auf Seite 187 beschrieben.
Namenslisten	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Namenslisten, wie in „Vollenden Verwaltungszugriff auf einige Namenslisten erteilen” auf Seite 187 beschrieben.
Services	Erteilen Sie vollständigen Verwaltungszugriff auf die erforderlichen Services, wie in „Vollenden Verwaltungszugriff auf einige Services erteilen” auf Seite 188 beschrieben.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie folgenden Befehle aus:

```
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. Unter z/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('QMgrNa
me')
```

- Geben Sie für z/OSdie folgenden Befehle aus:

```
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. Unter z/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie folgenden Befehle aus:

```
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. Unter z/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -t qmgr -g GroupName +alladm
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie folgenden Befehle aus:

```
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. Unter z/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
  ```

- Geben Sie für IBM iden folgenden Befehl aus:

  ```
  GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
  ```

- Geben Sie für z/OSdie folgenden Befehle aus:

  ```
  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. Unter z/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrNa
me')
```

- Geben Sie für z/OSdie folgenden Befehle aus:

```
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. Unter z/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie folgenden Befehle aus:

```
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. Unter z/OSkann 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.

Prozedur

- Mit dem Assistenten:
  a) Klicken Sie im Teilfenster WebSphere 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.
- Geben Sie für UNIX- und Windows -Systeme 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
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp +inq
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 MQ Explorer verwenden möchten.
- Geben Sie für IBM idie 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')
```

- Geben Sie für z/OSdie folgenden Befehle aus:

```
RDEFINE MQQUEUE QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQQUEUE) ID(GroupName) ACCESS(READ)
RDEFINE MQTOPIC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQTOPIC) 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. Unter z/OSkann 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

Verwenden Sie den Assistenten "Aufgabenbereichsbasierte Berechtigungen hinzufügen" oder die entsprechenden Befehle für Ihr Betriebssystem.

Prozedur

- Mit dem Assistenten:
  a) Klicken Sie im Teilfenster WebSphere 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.
- Geben Sie für UNIX and Linux -Systeme 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.QUEUE -t queue -g GroupName +dsp +inq +get
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 +conn
```

- Geben Sie für Windows-Systeme dieselben Befehle wie für UNIX and Linux -Systeme aus. Verwenden Sie jedoch den Profilnamen @CLASS anstelle von @class.
- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER('GroupName') AUT(*ALLADM) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie 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. Unter z/OSkann 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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName -connect
  ```

- Geben Sie für IBM iden folgenden Befehl aus:

  ```
  RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
  ```

- Geben Sie für z/OSdie 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. Unter z/OSkann 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 199 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 192 beschrieben gesichert werden.

In beiden Fällen muss die Verbindungssicherheit wie unter „Verbindungssicherheit einrichten” auf Seite 199 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 200.
Sie verfügen über Anwendungen, die Themen verwenden	Weitere Informationen finden Sie im Abschnitt „Benutzerzugriff auf Themen steuern” auf Seite 205.
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 206.
Sie verfügen über Anwendungen, die Prozessobjekte verwenden	Siehe „Zugriffsberechtigung für Zugriffsprozesse erteilen” auf Seite 207.
Sie verfügen über Anwendungen, die Namenslisten verwenden	Siehe „Berechtigung zum Zugriff auf Namenslisten erteilen” auf Seite 208.

Ferne Verbindung zum WS-Manager sichern

Zum Schutz von Fernverbindungen mit dem Warteschlangenmanager können SSL oder TLS, ein Sicherheitsexit, Kanalauthentifizierungsdatensätze oder eine Kombination aus all diesen Möglichkeiten eingesetzt werden.

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. Verwendung von SSL oder TLS mit Kanalauthentifizierungsdatensätzen:
   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. Verwendung von SSL oder TLS mit einem Sicherheitsexit:
   a) Setzen Sie MCAUSER auf dem Serververbindungskanal auf eine Benutzer-ID ohne Berechtigungen.
   b) Schreiben Sie einen Sicherheitsexit, der in Abhängigkeit vom Wert des SSL-DN, der dem Exit in der MQCD-Struktur in den Feldern "SSLPeerNamePtr" und "SSLPeerNameLength" übergeben wird, einen Wert für MCAUSER zuordnet.
3. Verwendung von SSL oder TLS mit festen Kanaldefinitionswerten:
   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. Verwendung von Kanalauthentifizierungsdatensätzen auf Kanälen, die kein SSL oder TLS 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 193
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 156. 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 unten angezeigten Befehlen angeben, hat dies 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

Bestätigte Client-Benutzer-ID einer MCAUSER-Benutzer-ID zuordnen
Über einen Kanalauthentifizierungssatz können Sie das Attribut MCAUSER eines Serververbindungskanals entsprechend der ursprünglichen Benutzer-ID 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) MCAU
SER(
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 steht für die vom Client bestätigte Benutzer-ID.
  user ist die Benutzer-ID, die anstelle des Clientbenutzernamens verwendet werden soll.

Zugehörige Verweise
SET CHLAUTH

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
) 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 WebSphere MQ-Standardregeln für SSLPEER-Werte folgt. Siehe WebSphere MQ -Regeln für SSLPEER-Werte.

user ist die Benutzer-ID, die für alle Verbindungen mit dem angegebenen DN verwendet werden soll.

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 dies 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

Zugriff für eine vom Client bestätigte Benutzer-ID blockieren

Über einen Kanalauthentifizierungssatz können Sie verhindern, dass eine bestätigte Client-Benutzer-ID Kanäle starten kann.

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(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.

client-user-name steht für die vom Client bestätigte Benutzer-ID.

Zugehörige Verweise

SET CHLAUTH

Zugriff durch einen definierten SSL-Namen blockieren

Sie können einen Kanalauthentifizierungsdatensatz verwenden, um zu verhindern, dass ein definierter SSL-Name Kanäle startet.

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')
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 WebSphere MQ-Standardregeln für SSLPEER-Werte folgt. Siehe WebSphere MQ -Regeln für SSLPEER-Werte.

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

- Löschen Sie alle Serververbindungskanäle mit dem 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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  ```

- Geben Sie für IBM iden folgenden Befehl aus:

  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM)  USER('GroupName') AUT(*CONNECT)
  ```

- Geben Sie für z/OSdie folgenden Befehle aus:

  ```
  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)
```

Diese Befehle erteilen die Berechtigung zum Herstellen einer Verbindung für Batch, CICS, IMS und den Kanalinitiator (CHIN). 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/OSkann 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.

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	Action
Die Anwendung ruft Nachrichten aus einer Warteschlange ab	Siehe „Erteilende Berechtigung zum Abrufen von Nachrichten aus Warteschlangen” auf Seite 200.
Die Anwendung definiert Kontext	Siehe „Berechtigung zum Festlegen des Kontexts erteilen” auf Seite 201.
Die Anwendung übergibt Kontext	Siehe „Berechtigung zum Übergeben des Kontexts erteilen” auf Seite 202.
Die Anwendung reiht Nachrichten in eine zu einem Cluster gehörigen Warteschlange ein	Siehe „Berechtigung zum Einreihen von Nachrichten in ferne Clusterwarteschlangen berechtigen” auf Seite 260.
Die Anwendung reiht Nachrichten in eine lokale Warteschlange ein	Siehe „Berechtigung zum Eingeben von Nachrichten in eine lokale Warteschlange erteilen” auf Seite 203.
Die Anwendung reiht Nachrichten in eine Modellwarteschlange ein	Siehe „Berechtigung zum Einreihen von Nachrichten in eine Modellwarteschlange erteilen” auf Seite 203.
Die Anwendung reiht Nachrichten in eine ferne Warteschlange ein	Siehe „Berechtigung zum Einlegen von Nachrichten in eine ferne Clusterwarteschlange erteilen” auf Seite 204.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie 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/OSkann 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.

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.

Prozedur

- Geben Sie unter UNIX, Linux und 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
```

- 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('QMgrName')
```

  - So legen Sie den gesamten Kontext fest:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('QMgrNa
me')
```

- Geben Sie für z/OSeine 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/OSkann 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.

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.

Prozedur

- Geben Sie unter UNIX, Linux und 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
```

- Geben Sie für IBM i einen der folgenden Befehle aus:
  - Nur Identitätskontext übergeben:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('QMgrNa
me')
```

  - So übergeben Sie den gesamten Kontext:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('QMgrNa
me')
```

- Setzen Sie für z/OSdie folgenden Befehle ab, 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/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie 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/OSkann 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.

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.

Prozedur

- Geben Sie unter UNIX, Linux und 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 idie 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/OSdie 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/OSkann dieser Wert auch der Name einer Gruppe mit gemeinsamer Warteschlange sein.

Modellwarteschlangenname
Der Name der Modellwarteschlange, auf der dynamische Warteschlangen basieren.

ObjectProfile
Der Name der dynamischen Warteschlange oder des generischen Profils, für die Berechtigungen geändert werden sollen.

GroupName
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 203. 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 spezifische Verhalten gilt nur, wenn Sie das Attribut **ClusterQueueAccessControl** in der Datei qm.ini als *RQMName* konfiguriert haben, wie in der Sicherheitszeilengruppe beschrieben, und den Warteschlangenmanager erneut gestartet haben.

Auf UNIX-, Linux-und Windows -Systemen können Sie auch den Befehl SET AUTHREC verwenden.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme 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 iden 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/OSdie folgenden Befehle aus:

```
RDEFINE MQQUEUE  QMgrNameObjectProfile UACC(NONE)
PERMIT QMgrNameObjectProfile CLASS(MQADMIN)
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/OSkann 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 16. Benutzerzugriff auf Themen steuern

Anweisung	**Action**
Die Anwendung veröffentlicht Nachrichten zu einem Thema	Siehe „Berechtigung zum Publizieren von Nachrichten in einem Thema erteilen” auf Seite 205.
Die Anwendung subskribiert ein Thema	Siehe „Berechtigung zum Subskribieren von Themen erteilen” auf Seite 206.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie 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/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie 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/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie 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/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME('QMgrName')
```

- Geben Sie für z/OSdie 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/OSkann 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.

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.

Prozedur

- Geben Sie für UNIX-, Linux -und Windows -Systeme den folgenden Befehl aus:

```
setmqaut -m QMgrName -n
ObjectProfile -t namelist -g GroupName
 +all
```

- Geben Sie für IBM iden folgenden Befehl aus:

```
GRTMQMAUT OBJ('ObjectProfile
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- Geben Sie für z/OSdie 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/OSkann 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.

Berechtigung zur Verwaltung von IBM WebSphere MQ auf UNIX, Linux, and Windows -Systemen

IBM WebSphere MQ-Administratoren können alle IBM WebSphere 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. Weitere Hinweise gelten für Windows -Systeme.

IBM WebSphere MQ -Administratoren haben die Berechtigung zur Verwendung aller WebSphere MQ -Befehle (einschließlich der Befehle zum Erteilen von WebSphere MQ -Berechtigungen für andere Benutzer)

Um ein IBM WebSphere MQ -Administrator zu sein, müssen Sie Mitglied einer speziellen Gruppe namens *mqm* sein (oder Mitglied der Administratorgruppe auf Windows -Systemen). Die Gruppe 'mqm' wird automatisch erstellt, wenn WebSphere MQ installiert wird. Fügen Sie weitere Benutzer zur Gruppe hinzu, damit sie die Verwaltung durchführen können. Alle Mitglieder dieser Gruppe haben Zugriff auf alle Ressourcen. dieser Zugriff kann nur entzogen werden, indem der betreffende Benutzer aus der Gruppe 'mqm' entfernt und der Befehl REFRESH SECURITY ausgegeben wird. Administratoren können Steuerbefehle verwenden, um WebSphere MQzu verwalten. Einer dieser Steuerbefehle ist **setmqaut**, mit dem anderen Benutzern Berechtigungen für den Zugriff auf oder die Steuerung von WebSphere MQ -Ressourcen erteilt werden. Die PCF-Befehle für die Verwaltung der Berechtigungssätze sind für Benutzer verfügbar, bei denen es sich nicht um Administratoren handelt, denen aber im Warteschlangenmanager die Berechti-

gungen 'dsp' und 'chg' erteilt wurden. Weitere Informationen zum Verwalten von Berechtigungen mithilfe von PCF-Befehlen finden Sie im Abschnitt Programmable Command Formats.

Administratoren können den Steuerbefehl **runmqsc** verwenden, um IBM WebSphere 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 einen Escape-PCF-Befehl eingebunden. Administratoren müssen über die erforderlichen Berechtigungen für die MQSC-Befehle verfügen, die vom fernen WS-Manager verarbeitet werden sollen. Der WebSphere MQ Explorer gibt PCF-Befehle aus, um Verwaltungstasks auszuführen. Administratoren benötigen keine zusätzlichen Berechtigungen für die Verwendung von WebSphere MQ Explorer, um einen WS-Manager auf dem lokalen System zu verwalten. Wenn der IBM WebSphere MQ Explorer verwendet wird, um einen Warteschlangenmanager auf einem anderen System zu verwalten, müssen Administratoren über die erforderlichen Berechtigungen verfügen, damit die PCF-Befehle von dem fernen Warteschlangenmanager verarbeitet werden können.

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 Authentifizierungsinformationsobjekte ausgeführt werden, finden Sie im Abschnitt Berechtigung für die Arbeit mit WebSphere 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

Außerdem hat das Konto SYSTEM auf Windows -Systemen uneingeschränkten Zugriff auf WebSphere MQ -Ressourcen.

Auf UNIX and Linux-Plattformen wird außerdem die Benutzer-ID 'mqm' erstellt, die nur vom Produkt verwendet werden soll. Es darf nie für nicht privilegierte Benutzer verfügbar sein. Eigner aller WebSphere MQ -Objekte ist die Benutzer-ID mqm.

Auf Windows -Systemen können Mitglieder der Administratorgruppe ebenso wie das Konto SYSTEM jeden Warteschlangenmanager verwalten. Sie können auch auf dem Domänencontroller eine mqm-Domänengruppe mit allen privilegierten Benutzer-IDs erstellen, die innerhalb der Domäne aktiv sind, und diese Gruppe zur lokalen mqm-Gruppe hinzufügen. Einige Befehle wie beispielsweise **crtmqm** bearbeiten Berechtigungen für IBM WebSphere 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, mit allen Objekten zu arbeiten, aber es kann Umstände auf Windows -Systemen geben, wenn die Berechtigung verweigert wird, wenn Sie einen lokalen Benutzer und einen domänenauthentifizierten Benutzer mit demselben Namen haben. Dieser Vorgang wird im Abschnitt „Principals und Gruppen” auf Seite 213 beschrieben.

Windows -Versionen mit einem UAC-Feature (User Account Control, Benutzerkontensteuerung) schränken die Aktionen ein, die Benutzer für bestimmte Betriebssystemfunktionen ausführen können, auch wenn sie Mitglieder der Administratorgruppe sind. Wenn Ihre Benutzer-ID zur Administratorgruppe, aber nicht zur Gruppe 'mqm' gehört, müssen Sie eine erweiterte Eingabeaufforderung verwenden, um WebSphere MQ -Verwaltungsbefehle wie **crtmqm**auszugeben. Andernfalls wird der Fehler "AMQ7077: Sie sind nicht berechtigt, die angeforderte Operation auszuführen" generiert. Um eine Eingabeaufforderung mit Administratorrechten zu öffnen, klicken Sie mit der rechten Maustaste auf das Startmenüelement oder Symbol für die Eingabeaufforderung und wählen Sie "Als Administrator ausführen" aus.

Sie müssen kein Mitglied der Gruppe mqm sein, um Folgendes zu tun:

- 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 75 beschrieben.)
- Geben Sie MQI-Aufrufe von einem Anwendungsprogramm aus aus (es sei denn, Sie möchten die Direktaufrufbindungen im Aufruf MQCONNX verwenden).

- Mit dem Befehl `crtmqcvx` ein Codefragment erstellen, mit dem für Datentypstrukturen eine Datenkonvertierung vorgenommen wird.
- Mit dem Befehl `dspmq` können Sie Warteschlangenmanager anzeigen.
- Zeigen Sie mit dem Befehl `dspmqtrc` formatierte WebSphere MQ-Traceausgaben an.

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 Version 5.3 hat diesen Grenzwert erhöht, aber WebSphere MQ unterliegt weiterhin einer Beschränkung von 12 Zeichen auf allen UNIX and Linux -Plattformen. Wenn Sie eine Benutzer-ID mit mehr als 12 Zeichen verwenden, ersetzt WebSphere MQ sie durch den Wert `UNKNOWN`. Definieren Sie keine Benutzer-ID mit dem Wert `UNKNOWN`.

Gruppe 'mqm' verwalten

Benutzer in der Gruppe 'mqm' erhalten vollständige Administratorberechtigungen für WebSphere MQ. Aus diesem Grund sollten Sie keine Anwendungen und normalen Benutzer in der Gruppe 'mqm' registrieren. Die Gruppe mqm sollte nur die Konten der WebSphere MQ -Administratoren enthalten.

Diese Tasks werden in beschrieben:

- Gruppen unter Windows erstellen und verwalten
- Gruppen unter HP-UX erstellen und verwalten
- Gruppen unter AIX erstellen und verwalten
- Gruppen unter Solaris erstellen und verwalten
- Gruppen unter Linux erstellen und verwalten

Wenn Ihr Domänencontroller unter Windows 2000 oder Windows 2003 ausgeführt wird, muss Ihr Domänenadministrator möglicherweise ein spezielles Konto für die Verwendung durch WebSphere MQ einrichten. Dies wird im Abschnitt WebSphere MQ -Konten konfigurierenbeschrieben.

Berechtigung zum Arbeiten mit IBM WebSphere MQ -Objekten auf UNIX, Linux, and Windows -Systemen

Alle Objekte werden durch IBM WebSphere MQ geschützt, und Principals müssen über die entsprechende Berechtigung zum Zugriff auf diese Objekte berechtigt werden. 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 WebSphere MQgeschützt, und Anwendungen müssen die Berechtigung für den Zugriff auf sie erhalten. 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. Beispielsweise kann eine Gruppe für eine bestimmte Warteschlange sowohl Put-als auch Get-Operationen ausführen; eine andere Gruppe kann nur die Warteschlange durchsuchen (`MQGET` mit Suchoption). 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 erteilt, die das Objekt erstellt hat, sowie allen Mitgliedern der Gruppe mqm (und den Mitgliedern der lokalen Administratorgruppe auf Windows -Systemen).

Zugehörige Konzepte

„Berechtigung zur Verwaltung von IBM WebSphere MQ auf UNIX, Linux, and Windows -Systemen" auf Seite 208

IBM WebSphere MQ-Administratoren können alle IBM WebSphere 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. Weitere Hinweise gelten für Windows -Systeme.

Bei Sicherheitsprüfungen auf UNIX, Linux, and Windows -Systemen

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. WebSphere MQ prüft dann, ob die Benutzer-ID berechtigt ist, eine Verbindung zum WS-Manager herzustellen, und behält die Benutzer-ID für zukünftige Prüfungen bei.

Benutzer müssen sich nicht bei WebSphere MQanmelden; WebSphere MQ geht davon aus, dass Benutzer sich am zugrunde liegenden Betriebssystem angemeldet haben und von diesem authentifiziert wurden.

Das Objekt öffnen (MQOPEN-oder MQPUT1-Aufrufe)
Auf WebSphere MQ -Objekte wird zugegriffen, indem das Objekt geöffnet und Befehle für das Objekt abgesetzt 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).

WebSphere MQ überprüft die Ressource, die in der Anforderung **MQOPEN** 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. Dies wird in Anmerkung „1" auf Seite 96 beschrieben.

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 ausgeführt, die in der Themenstruktur an oder oberhalb des Punkts in der Themenstruktur gefunden werden, an dem 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.

Implementierung der Zugriffssteuerung durch IBM WebSphere MQ auf UNIX, Linux, and Windows -Systemen

IBM WebSphere MQ verwendet die vom zugrunde liegenden Betriebssystem bereitgestellten Sicherheitsservice mit dem Objektberechtigungsmanager. IBM WebSphere MQ stellt Befehle bereit, mit denen Zugriffssteuerungslisten erstellt und verwaltet werden.

Eine Zugriffssteuerungsschnittstelle, die als Berechtigungsserviceschnittstelle bezeichnet wird, ist Teil von WebSphere MQ. WebSphere MQ stellt eine Implementierung eines Zugriffssteuerungsmanagers (entsprechend der Berechtigungsserviceschnittstelle) bereit, der als *Object Authority Manager (OAM)*bezeichnet wird. Diese Option wird automatisch für jeden von Ihnen erstellten Warteschlangenmanager installiert und aktiviert, sofern Sie nichts anderes angeben (wie in „Sicherheitszugriffsprüfungen auf Systemen mit UNIX, Linux, and Windows verhindern” auf Seite 177 beschrieben). 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 WebSphere MQ -Objekte zugreifen, wenn sie die richtige Berechtigung haben. Im Abschnitt „Zugriff auf Objekte über OAM auf UNIX-, Linux -und Windows -Systemen steuern” auf Seite 169 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 beschränkt, die mit der ID SYSTEM angemeldet sind. Der Benutzerzugriff auf die Warteschlange kann nicht geändert werden.

WebSphere MQ stellt Befehle zum Erstellen und Verwalten von Zugriffssteuerungslisten bereit. Weitere Informationen zu diesen Befehlen finden Sie im Abschnitt „Zugriff auf Objekte über OAM auf UNIX-, Linux -und Windows -Systemen steuern” auf Seite 169.

WebSphere MQ übergibt dem OAM eine Anforderung, die einen Principal, einen Ressourcennamen und einen Zugriffstyp enthält. Der OAM erteilt oder verweigert den Zugriff auf der Basis der ACL, die er verwaltet. WebSphere MQ folgt der Entscheidung des OAM. Wenn der OAM keine Entscheidung treffen kann, lässt WebSphere MQ keinen Zugriff zu.

Benutzer-ID auf UNIX, Linux, and Windows -Systemen 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 WebSphere 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 UNIX 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 WebSphere MQ gibt die Benutzer-ID, die vom System im Nachrichtenheader (MQMD-Struktur) von jeder Nachricht empfangen wird, als die Kennung des Benutzers weiter. Diese Kennung ist Teil der Nachrichtenkontextinformationen und wird in „Kontextberechtigung auf UNIX-, Linux -und Windows -Systemen” auf Seite 215 beschrieben. Anwendungen können diese Informationen nur ändern, wenn sie zum Ändern von Kontextinformationen berechtigt sind.

Principals und Gruppen

Principals können zu Gruppen gehören. Sie können Gruppen statt Einzelpersonen Zugriff auf eine bestimmte Ressource erteilen, um den erforderlichen Verwaltungsaufwand zu reduzieren. Auf UNIX and Linux -Systemen basieren alle Zugriffssteuerungslisten (ACLs) auf Gruppen, auf Windows -Systemen jedoch auf Benutzer-IDs und Gruppen.

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 wird in beschrieben:

- Gruppen unter Windows erstellen und verwalten
- Gruppen unter HP-UX erstellen und verwalten
- Gruppen unter AIX erstellen und verwalten
- Gruppen unter Solaris erstellen und verwalten
- Gruppen unter Linux 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 das PCF-Äquivalent) aus.

UNIX and Linux-Systeme

Alle ACLs basieren auf Gruppen. Wenn einem Benutzer Zugriff auf eine bestimmte Ressource erteilt wird, wird die Primärgruppe der Benutzer-ID in die ACL 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 Zugriff auf WebSphere MQ -Ressourcen zu erteilen.

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.

Benutzer-IDs und Gruppennamen können bis zu 12 Zeichen enthalten.

Windows-Systeme

ACLs basieren sowohl auf Benutzer-IDs als auch auf Gruppen. Prüfungen sind mit denen für UNIX -Systeme identisch, außer dass einzelne Benutzer-IDs auch in der ACL angezeigt werden können. Sie können unterschiedliche Benutzer in verschiedenen Domänen mit derselben Benutzer-ID haben. WebSphere MQ ermöglicht die Qualifizierung von Benutzer-IDs durch einen Domänennamen, sodass diesen Benutzern unterschiedliche Zugriffsebenen zugeordnet werden können.

Der Gruppenname kann optional einen Domänennamen enthalten, der in den folgenden Formaten angegeben wird:

```
GroupName@domain
domain\GroupName
```

Globale Gruppen werden vom OAM nur in zwei Fällen überprüft:

1. Die Sicherheitszeilengruppe des Warteschlangenmanagers enthält die Einstellung `GroupModel=GlobalGroups`; siehe Sicherheit.
2. Der Warteschlangenmanager 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.

Einige Steuerbefehle (z. B. `crtmqm`) ändern Berechtigungen für WebSphere MQ -Objekte mithilfe des Objektberechtigungsmanagers (OAM). 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.

Windows -Sicherheitskennungen (SIDs)

WebSphere MQ unter Windows verwendet die SID, unter der sie verfügbar ist. Wenn eine Windows -SID nicht mit einer Berechtigungsanforderung bereitgestellt wird, identifiziert WebSphere MQ den Benutzer allein anhand des Benutzernamens. Dies kann jedoch dazu führen, dass die falsche Berechtigung erteilt wird.

Auf Windows -Systemen wird die Sicherheits-ID (SID) als Ergänzung zur Benutzer-ID verwendet. Die SID enthält Informationen, die die vollständigen Benutzerkontodetails in der Windows -Sicherheitskontenmanagerdatenbank (SAM) angeben, in der der Benutzer definiert ist. Wenn eine Nachricht in WebSphere MQ für Windowserstellt wird, speichert WebSphere MQ die SID im Nachrichtendeskriptor. Wenn WebSphere MQ unter Windows Berechtigungsprüfungen ausführt, verwendet es die SID zum Abfragen der vollständigen Informationen aus der SAM-Datenbank. (Die SAM-Datenbank, in der der Benutzer definiert ist, muss zugänglich sein, damit diese Abfrage erfolgreich ausgeführt werden kann.)

Wenn keine Windows -SID mit einer Berechtigungsanforderung bereitgestellt wird, identifiziert WebSphere MQ den Benutzer standardmäßig allein anhand 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 die falsche WebSphere MQ -Berechtigung erteilt. Um dieses Problem zu vermeiden, schließen Sie in jede Berechtigungsanfrage eine SID ein. Die SID wird von WebSphere MQ verwendet, um Benutzerberechtigungsnachweise einzurichten.

Um anzugeben, dass alle Berechtigungsanforderungen eine SID enthalten müssen, verwenden Sie `regedit`. Setzen Sie die Sicherheitsrichtlinie auf NTSIDsRequired.

Alternative Benutzerberechtigung auf UNIX-, Linux -und Windows -Systemen

Sie können angeben, dass eine Benutzer-ID die Berechtigung eines anderen Benutzers beim Zugriff auf ein WebSphere MQ -Objekt verwenden kann. Dies wird als *alternative Benutzerberechtigung*bezeichnet und kann für jedes WebSphere MQ -Objekt verwendet werden.

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.

Kontextberechtigung auf UNIX-, Linux -und Windows -Systemen

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 entweder bei einem `MQOPEN`- oder einem `MQPUT`-Aufruf angeben.

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 entweder bei einem `MQOPEN`- oder einem `MQPUT`-Aufruf angeben. 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. Mit der Kontextberechtigung können Sie steuern, ob der Benutzer eine der Kontextoptionen in einem `MQOPEN` -oder `MQPUT1` -Aufruf 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 WebSphere 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. *MCAUserIdentifier* enthält die ersten 12 Byte der MCA-Benutzer-ID. Wenn die MCA-Benutzer-ID nicht leer ist, gibt sie die Benutzer-ID an, die vom Nachrichtenkanalagenten für die Zugriffsberechtigung auf MQ -Ressourcen verwendet werden soll. Stellen Sie sicher, dass MCAUSER auf Windows-Plattformen weniger als 12 Zeichen umfasst.

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.

Auf IBM i-, UNIX-, Linux -und Windows -Systemen verwendet der Warteschlangenmanager den Wert von *MCAUserIdentifier* als Benutzer-ID für Berechtigungsprüfungen, wenn ein MCA versucht, auf die Ressourcen des Warteschlangenmanagers zuzugreifen, nachdem er mit dem Warteschlangenmanager verbunden wurde, es sei denn, der Wert von *MCAUserIdentifier* ist leer. 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.

Unter z/OSverwendet der Warteschlangenmanager möglicherweise den Wert von *MCAUserIdentifier* für Berechtigungsprüfungen, sofern dieser 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:

- Wenn am Clientende eines MQI-Kanals kein Sicherheitsexit vorhanden ist, fließt eine Benutzer-ID, die der WebSphere MQ -Clientanwendung zugeordnet ist, von der Clientverbindung MCA zur Serververbindung MCA, 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. In einem MQI-Kanal kann ein Sicherheitsexit auf der Clientseite

des Kanals die Benutzer-ID senden, die der WebSphere MQ MQI-Clientanwendung 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.

- Wird auf dem Kanal SSL verwendet, wird der definierte Name (DN) des Partners an den Exit im Feld 'SSLPeerNamePtr' der MQCD-Struktur übergeben. Der definierte Name des Ausstellers des Zertifikats wird an den Exit im Feld 'SSLRemCertIssNamePtr' der MQCXP-Struktur 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 WebSphere MQ 7.1 erstellt wurden, müssen unter Umständen aktualisiert werden. Weitere Informationen finden Sie im Abschnitt Kanalsicherheits-Exitprogramme.

Benutzerauthentifizierung für WebSphere MQ -Objektberechtigungsmanager

Unter WebSphere MQ MQI-Clientverbindungen können Sicherheitsexits verwendet werden, um die MQCSP-Struktur zu ändern oder zu erstellen, die in der OAM-Benutzerauthentifizierung (Object Authority Manager) 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 bitten, diese Benutzer-ID für Berechtigungsprüfungen zu verwenden, wenn er für den Client auf WebSphere MQ -Ressourcen zugreift.

Wenn der Parameter PUTAUT in der Kanaldefinition auf CTX (oder ALTMCA unter z/OS) gesetzt ist, wird die Benutzer-ID im Feld *UserIdentifier* jeder eingehenden Nachricht für Berechtigungsprüfungen verwendet, wenn der MCA 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 in „Identitätsabgleich in Nachrichtenexits” auf Seite 158.

Zugriffssteuerung in API-Exit und API-Steuerübergabeexit implementieren

Ein API-oder API-Steuerübergabeexit kann Zugriffssteuerungen bereitstellen, um die von WebSphere MQbereitgestellten zu 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.

Vertraulichkeit von Nachrichten

Um die Vertraulichkeit zu wahren, verschlüsseln Sie Ihre Nachrichten. Je nach Bedarf gibt es verschiedene Methoden zum Verschlüsseln von Nachrichten in WebSphere MQ .

Ihre Wahl von CipherSpec bestimmt, welches Maß an Vertraulichkeit Sie haben.

Wenn Sie einen End-to-End-Datenschutz auf Anwendungsebene für Ihre Punkt-zu-Punkt-Messaging-Infrastruktur benötigen, können Sie die Nachrichten mit WebSphere MQ Advanced Message Security verschlüsseln oder einen eigenen API-Exit oder API-Steuerübergabeexit schreiben.

Wenn Sie Nachrichten nur während des Transports durch einen Kanal verschlüsseln müssen, da Sie auf Ihren Warteschlangenmanagern über eine ausreichende Sicherheit verfügen, können Sie SSL oder TLS verwenden, oder Sie können einen eigenen Sicherheitsexit, Nachrichtenexit oder eigene Sende- und Empfangsexitprogramme schreiben.

Weitere Informationen zu WebSphere MQ Advanced Message Securityfinden Sie unter „ Erweiterte Nachrichtensicherheit-Planung” auf Seite 68.Die Verwendung von SSL und TLS mit WebSphere MQ wird unter „IBM WebSphere MQ Unterstützung für SSL und TLS” auf Seite 25beschrieben. Die Verwendung von Exitprogrammen in der Nachrichtenverschlüsselung wird unter „Vertraulichkeit in Benutzerexitprogrammen implementieren” auf Seite 238 beschrieben.

Zwei Warteschlangenmanager über SSL oder TLS verbinden

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. Zudem müssen Sie Ihre digitalen Zertifikate anfordern und verwalten. Auf einem Testsystem können Sie selbst signierte Zertifikate verwenden, die von einer lokalen Zertifizierungsstelle ausgegeben wurden. Verwenden Sie selbst signierte Zertifikate nicht auf einem Produktionssystem. Weitere Informationen finden Sie unter ..//zs14140_.dita.

Vollständige Informationen zum Erstellen und Verwalten von Zertifikaten finden Sie in „Mit SSL oder TLS auf UNIX, Linux, and Windows -Systemen arbeiten” auf Seite 120.

In diesen Themen werden die Aufgaben bei der Einrichtung der SSL-Kommunikation erläutert, zudem erhalten Sie dort eine schrittweise Anleitung zum Ausführen dieser Aufgaben.

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 WebSphere MQ fordert der SSL-oder TLS-Server immer ein Zertifikat vom Client an.

Anmerkungen:

1. In diesem Kontext bezieht sich ein SSL-Client auf die Verbindung, die den Handshake initialisiert.
2. Weitere Details finden Sie im Glossar .

Auf UNIX-, Linux -und Windows -Systemen sendet der SSL-oder TLS-Client nur dann ein Zertifikat, wenn es über ein Zertifikat im richtigen WebSphere MQ -Format verfügt, das `ibmwebspheremq` gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben lautet. Beispiel: `ibmwebspheremqqm1` für `QM1`.

WebSphere MQ verwendet das Präfix `ibmwebspheremq` in einem Kennsatz, um Verwechslungen 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. Sendet der Client kein Zertifikat, schlägt die Authentifizierung nur dann fehl, wenn der Parameter SSLCAUTH für das Ende des Kanals, das als SSL- oder TLS-Server agiert, auf REQUIRED gesetzt ist oder ein Wert für den Parameter SSLPEER angegeben ist. Weitere Informationen zur anonymen Verbindung eines Warteschlangenmanagers (d. h. wenn der SSL- oder TLS-Client kein Zertifikat sendet) finden Sie im Abschnitt „Zwei Warteschlangenmanager über unidirektionale Authentifizierung verbinden” auf Seite 223.

Selbst signierte Zertifikate für die gegenseitige Authentifizierung zweier Warteschlangenmanager verwenden

Folgen Sie diesen Beispielanweisungen, um die gegenseitige Authentifizierung zweier Warteschlangenmanager mithilfe von selbst signierten SSL- oder TLS-Zertifikaten zu implementieren.

Informationen zu diesem Vorgang

Szenario:

- Sie haben zwei Warteschlangenmanager, QM1 und QM2, die sicher miteinander kommunizieren müssen. Deswegen sollen sich QM1 und QM2 gegenseitig authentifizieren.
- Die sichere Kommunikation möchten Sie mit selbst signierten Zertifikaten testen.

Die Konfiguration wird danach so aussehen:

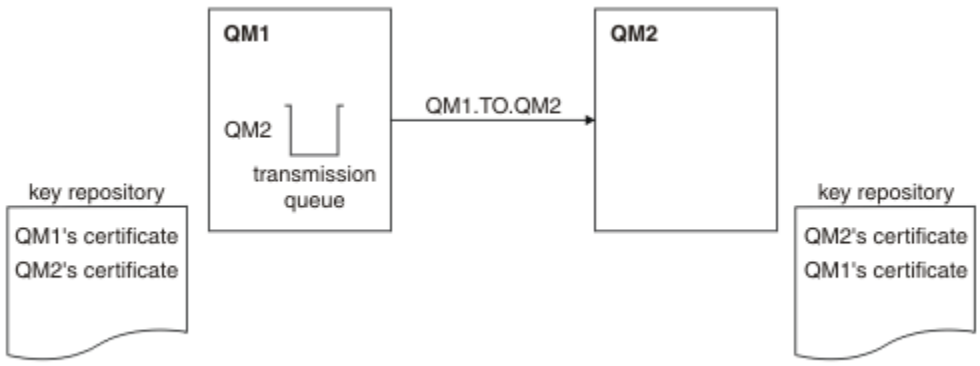


Abbildung 14. Aus dieser Task entstehende Konfiguration

In Abbildung 14 auf Seite 219 enthält das Schlüsselrepository von QM1 das Zertifikat für QM1 und das öffentliche Zertifikat von QM2. Das Schlüsselrepository von QM2 enthält das Zertifikat für QM2 und das öffentliche Zertifikat von QM1.

Vorgehensweise

1. Bereiten Sie das Schlüsselrepository auf beiden Warteschlangenmanagern entsprechend des Betriebssystems vor:
   - Auf UNIX-, Linux-und Windows -Systemen.
2. Erstellen Sie für jeden Warteschlangenmanager ein selbst signiertes Zertifikat:
   - Auf UNIX-, Linux-und Windows -Systemen.
3. Extrahieren Sie eine Kopie jedes Zertifikats:
   - Auf UNIX-, Linux-und Windows -Systemen.
4. Übertragen Sie den öffentlichen Teil des Zertifikats QM1 mit einem Dienstprogramm wie FTP).
5. Fügen Sie auf jedem Warteschlangenmanager das Partnerzertifikat zum Schlüsselrepository hinzu:
   - Auf UNIX-, Linux-und Windows -Systemen.
6. Definieren Sie auf QM1 einen Senderkanal und die zugehörige Übertragungswarteschlange, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
   SSLCIPH(RC4_MD5_US) DESCR('Sender channel using SSL from QM1 to QM2')

   DEFINE QLOCAL(QM2) USAGE(XMITQ)
   ```

   Dieses Beispiel verwendet die CipherSpec RC4_MD5. Die CipherSpecs an jedem Ende des Kanals müssen identisch sein.
7. Definieren Sie auf QM2 einen Empfängerkanal, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from QM1 to QM2')
   ```

   Der Kanal muss denselben Namen wie der in Schritt 6 definierte Senderkanal haben und die gleiche CipherSpec verwenden.
8. Starten Sie den Kanalbeschrieben.

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden erstellt, wie in Abbildung 14 auf Seite 219 gezeigt.

Nächste Schritte

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe von DISPLAY-Befehlen. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie in den folgenden Beispielen aus.

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM1.TO.QM2)                     CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

Geben Sie auf dem Warteschlangenmanager QM2 den folgenden Befehl ein:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM2.TO.QM1)                     CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QM1)
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

In jedem Fall muss der Wert von SSLPEER mit dem DN im Partnerzertifikat übereinstimmen, das in Schritt 2 erstellt wurde. Da es sich um ein selbst signiertes Zertifikat handelt, entspricht der Name des Ausstellers dem Namen der Partnerwarteschlange.

SSLPEER ist optional. Wenn dieses Attribut allerdings angegeben ist, muss sein Wert den DN des in Schritt 2 erstellten Partnerzertifikat zulassen. Weitere Informationen zur Verwendung von SSLPEER finden Sie unter WebSphere MQ -Regeln für SSLPEER-Werte .

Von Zertifizierungsstelle signierte Zertifikate für die gegenseitige Authentifizierung zweier Warteschlangenmanager verwenden

Folgen Sie diesen Beispielanweisungen, um die gegenseitige Authentifizierung zweier Warteschlangenmanager mithilfe von SSL- oder TLS-Zertifikaten einer Zertifizierungsstelle zu implementieren.

Informationen zu diesem Vorgang

Szenario:

- Sie haben zwei Warteschlangenmanager, QMA und QMB, die sicher miteinander kommunizieren müssen. Deswegen sollen sich QMA und QMB gegenseitig authentifizieren.
- Dieses Netz wollen Sie später in einer Produktionsumgebung verwenden, weshalb Sie von Anfang an CA-signierte Zertifikate verwenden möchten.

Die Konfiguration wird danach so aussehen:

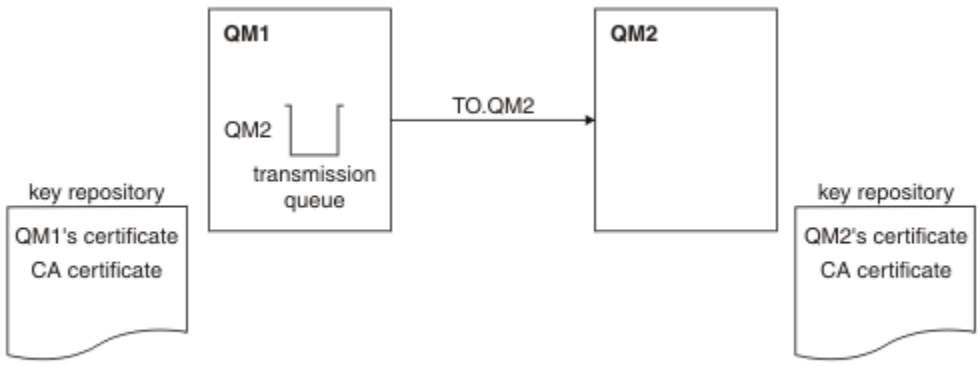


Abbildung 15. Aus dieser Task entstehende Konfiguration

In Abbildung 15 auf Seite 221 enthält das Schlüsselrepository von QMA das Zertifikat für QMA sowie das Zertifikat der Zertifizierungsstelle. Das Schlüsselrepository von QMB enthält das Zertifikat für QMB und das Zertifikat der Zertifizierungsstelle. In diesem Beispiel wurden die Zertifikate von QMA und QMB von der gleichen Zertifizierungsstelle ausgestellt. Wenn die beiden CA-Zertifikate von zwei verschiedenen Zertifizierungsstellen ausgestellt worden wären, müssten die Schlüsselrepositorys von QMA und QMB beide Zertifikate enthalten.

Vorgehensweise

1. Bereiten Sie das Schlüsselrepository auf beiden Warteschlangenmanagern entsprechend des Betriebssystems vor:
   - Auf UNIX-, Linux-und Windows -Systemen.
2. Fordern Sie für jeden Warteschlangenmanager ein von einer Zertifizierungsstelle signiertes Zertifikat an.

   Sie können die Zertifikate von zwei verschiedenen Zertifizierungsstellen anfordern.
   - Auf UNIX-, Linux-und Windows -Systemen.
3. Fügen Sie das CA-Zertifikat zum Schlüsselrepository des jeweiligen Warteschlangenmanagers hinzu:

   Wenn Sie für die Warteschlangenmanager verschiedene Zertifizierungsstellen verwenden, müssen Sie das CA-Zertifikat jeder Zertifizierungsstelle beiden Schlüsselrepositorys hinzufügen.
   - Auf UNIX-, Linux-und Windows -Systemen.
4. Fügen Sie auf jedem Warteschlangenmanager das von der Zertifizierungsstelle signierte Zertifikat zum Schlüsselrepository hinzu:
   - Auf UNIX-, Linux-und Windows -Systemen.
5. Definieren Sie auf QMA einen Senderkanal und die zugehörige Übertragungswarteschlange, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
   SSLCIPH(RC2_MD5_EXPORT) DESCR('Sender channel using SSL from QMA to QMB')

   DEFINE QLOCAL(QMB) USAGE(XMITQ)
   ```

   Dieses Beispiel verwendet die CipherSpec RC4_MD5. Die CipherSpecs an jedem Ende des Kanals müssen identisch sein.
6. Definieren Sie auf QMB einen Empfängerkanal, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL to QMB')
   ```

   Der Kanal muss denselben Namen wie der in Schritt 6 definierte Senderkanal haben und die gleiche CipherSpec verwenden.
7. Starten Sie den Kanal:

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden erstellt, wie in Abbildung 15 auf Seite 221 gezeigt.

Nächste Schritte

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe von DISPLAY-Befehlen. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie in den folgenden Beispielen aus.

Geben Sie auf dem Warteschlangenmanager QMA den folgenden Befehl ein:

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QMB)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
```

```
   XMITQ(QMB)
```

Geben Sie auf dem Warteschlangenmanager QMB den folgenden Befehl ein:

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

In jedem Fall muss der Wert von SSLPEER mit dem Wert des definierten Namens (DN) im Partnerzertifikat übereinstimmen, der in Schritt 2 erstellt wurde. Der Name des Ausstellers entspricht dem DN des CA-Zertifikats, das das persönliche Zertifikat, das in Schritt 4 hinzugefügt wurde, signiert hat.

Zwei Warteschlangenmanager über unidirektionale Authentifizierung verbinden

Folgen Sie diesen Beispielanweisungen, um ein System mit gegenseitiger Authentifizierung so zu ändern, dass sich Warteschlangenmanager über die unidirektionale Authentifizierung miteinander verbinden können (für den Fall, dass der SSL- oder TLS-Client kein Zertifikat sendet).

Informationen zu diesem Vorgang

Szenario:

- Ihre beiden Warteschlangenmanager (QM1 und QM2) sind eingerichtet, wie in „Von Zertifizierungsstelle signierte Zertifikate für die gegenseitige Authentifizierung zweier Warteschlangenmanager verwenden" auf Seite 221 beschrieben.
- Sie möchten QM1 so ändern, dass er sich über die unidirektionale Authentifizierung mit QM2 verbinden kann.

Die Konfiguration wird danach so aussehen:

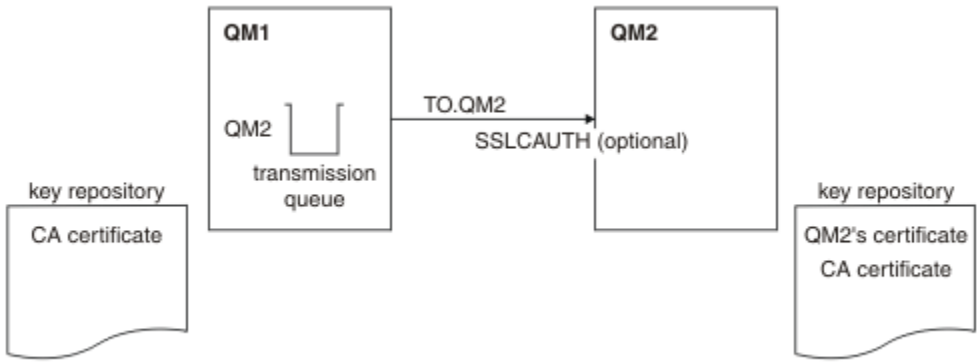


Abbildung 16. Warteschlangenmanager lassen unidirektionale Authentifizierung zu

Vorgehensweise

1. Entfernen Sie je nach Betriebssystem das persönliche Zertifikat von QM1aus dem zugehörigen Schlüsselrepository:
   - Auf UNIX-, Linux-und Windows -Systemen. Das Zertifikat ist wie folgt gekennzeichnet:
     – `ibmwebspheremq` gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben. Beispiel für `QM1` : `ibmwebspheremqqm1`.
2. Optional: Aktualisieren Sie auf QM1, wenn zuvor SSL-oder TLS-Kanäle ausgeführt wurden, die SSL- oder TLS-Umgebungbeschrieben.
3. Erlauben Sie anonyme Verbindungen auf dem Empfängerbeschrieben.

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden geändert, wie in Abbildung 16 auf Seite 223 gezeigt.

Nächste Schritte

Wenn der Senderkanal aktiv war und Sie in Schritt 2 den Befehl REFRESH SECURITY TYPE(SSL) ausgegeben haben, wird der Kanal automatisch gestartet. Wenn der Senderkanal inaktiv war, starten Sie ihn.

Das Vorhandensein des Parameterwerts für den Peer-Namen im Kanalstatus am Serverende des Kanals ist ein Hinweis darauf, dass ein Clientzertifikat übertragen wurde.

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe einiger DISPLAY-Befehle. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie in den folgenden Beispielen aus:

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

Geben Sie auf dem Warteschlangenmanager QM2 den folgenden Befehl ein:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

Auf QM2 ist das Feld SSLPEER leer, da von QM1 kein Zertifikat gesendet wurde. Auf QM1 stimmt der Wert von SSLPEER mit dem DN im persönlichen Zertifikat von QM2 überein.

Client sicher mit einem WS-Manager verbinden

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. Zudem müssen Sie Ihre digitalen Zertifikate anfordern und verwalten. Auf einem Testsystem können Sie selbst signierte Zertifikate verwenden, die von einer lokalen Zertifizierungsstelle ausgegeben wurden. Verwenden Sie selbst signierte Zertifikate nicht auf einem Produktionssystem. Weitere Informationen finden Sie unter ..//zs14140_.dita.

Vollständige Informationen zum Erstellen und Verwalten von Zertifikaten finden Sie in „Mit SSL oder TLS auf UNIX, Linux, and Windows -Systemen arbeiten” auf Seite 120.

In diesen Themen werden die Aufgaben bei der Einrichtung der SSL-Kommunikation erläutert, zudem erhalten Sie dort eine schrittweise Anleitung zum Ausführen dieser Aufgaben.

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 WebSphere MQ fordert der SSL-oder TLS-Server immer ein Zertifikat vom Client an.

Auf UNIX, Linux, and Windows -Systemen sendet der SSL-oder TLS-Client ein Zertifikat nur, wenn es über ein Zertifikat verfügt, das im richtigen WebSphere MQ -Format beschriftet ist. Auf dieses Format folgt `ibmwebspheremq` , auf das Ihre Anmelde-Benutzer-ID in Kleinbuchstaben geändert wurde, z. B. `ibmwebspheremqmyuserid`.

WebSphere MQ verwendet das Präfix `ibmwebspheremq` in einem Kennsatz, um Verwechslungen 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. Sendet der Client kein Zertifikat, schlägt die Authentifizierung nur dann fehl, wenn der Parameter SSLCAUTH für das Ende des Kanals, das als SSL- oder TLS-Server agiert, auf REQUIRED gesetzt ist oder ein Wert für den Parameter SSLPEER angegeben ist. Weitere Informationen zur anonymen Verbindung eines Warteschlangenmanagers finden Sie unter „Client anonym mit einem Warteschlangenmanager verbinden” auf Seite 229.

Selbst signierte Zertifikate für die gegenseitige Authentifizierung von Client und Warteschlangenmanager verwenden

Folgen Sie diesen Beispielanweisungen, um die gegenseitige Authentifizierung zwischen einem Client und einem Warteschlangenmanager mithilfe von selbst signierten SSL- oder TLS-Zertifikaten zu implementieren.

Informationen zu diesem Vorgang

Szenario:

- Sie haben Client C1 und Warteschlangenmanager QM1, die sicher miteinander kommunizieren müssen. Deswegen sollen sich C1 und QM1 gegenseitig authentifizieren.
- Die sichere Kommunikation möchten Sie mit selbst signierten Zertifikaten testen.

Unter IBM i unterstützt DCM keine selbst signierten Zertifikate, weshalb diese Task auf IBM i-Systemen nicht anwendbar ist.

Die Konfiguration wird danach so aussehen:

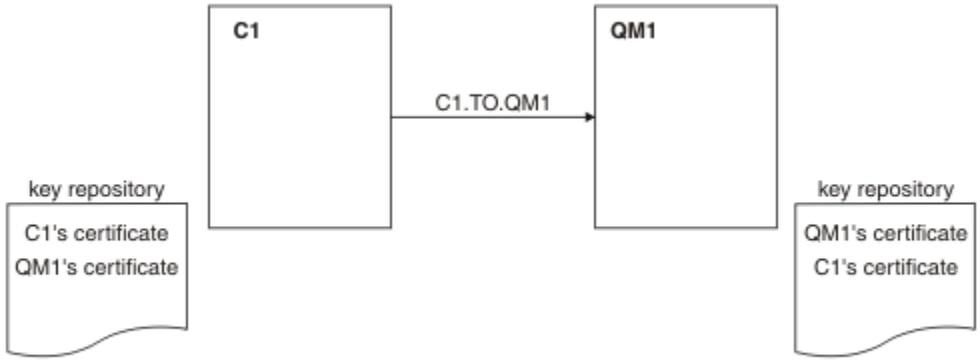


Abbildung 17. Aus dieser Task entstehende Konfiguration

In Abbildung 17 auf Seite 226 enthält das Schlüsselrepository von QM1 das Zertifikat für QM1 und das öffentliche Zertifikat von C1. Das Schlüsselrepository von C1 enthält das Zertifikat für C1 und das öffentliche Zertifikat von QM1.

Vorgehensweise

1. Bereiten Sie das Schlüsselrepository auf dem Client und dem Warteschlangenmanager entsprechend des Betriebssystems vor:
   - Auf UNIX-, Linux-und Windows -Systemen.
2. Erstellen Sie für den Client und den Warteschlangenmanager ein selbst signiertes Zertifikat:
   - Auf UNIX-, Linux-und Windows -Systemen.
3. Extrahieren Sie eine Kopie jedes Zertifikats:
   - Auf UNIX-, Linux-und Windows -Systemen.
4. Übertragen Sie den öffentlichen Teil des Zertifikats C1 auf das System QM1 und umgekehrt, indem Sie ein Dienstprogramm wie FTPbeschrieben.
5. Fügen Sie auf dem Client und dem Warteschlangenmanager jeweils das Partnerzertifikat zum Schlüsselrepository hinzu:
   - Auf UNIX-, Linux-und Windows -Systemen.
6. Geben Sie den Befehl REFRESH SECURITY TYPE(SSL) auf dem Warteschlangenmanager aus.
7. Definieren Sie mit einer der folgenden Methoden einen Clientverbindungskanal:
   - Verwendung des MQCONNX-Aufrufs mit der MQSCO-Struktur auf C1, wie unter Clientverbindungskanal auf dem WebSphere MQ MQI-Client erstellenbeschrieben.
   - Verwendung einer Definitionstabelle für Clientkanäle, wie unter Serververbindungs-und Clientverbindungsdefinitionen auf dem Server erstellen beschrieben.
8. Definieren Sie auf QM1 einen Serververbindungskanal, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   Der Kanal muss denselben Namen wie der in Schritt 6 definierte Clientverbindungskanal haben und die gleiche CipherSpec verwenden.

Ergebnisse

Schlüsselrepositorys und Kanäle werden wie in Abbildung 17 auf Seite 226 dargestellt erstellt.

Nächste Schritte

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe von DISPLAY-Befehlen. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie im folgenden Beispiel aus.

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                       CURRENT
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                           SUBSTATE(RECEIVE)
```

Optional können Sie für die Kanaldefinitionen auch das Filterattribut SSLPEER festlegen. Wenn die Kanaldefinition SSLPEER festgelegt ist, muss ihr Wert mit dem registrierten Namen des Zertifikatsinhabers im Partnerzertifikat übereinstimmen, das in Schritt 2 erstellt wurde. Nach einer erfolgreichen Verbindung zeigt das Feld SSLPEER in der Ausgabe von DISPLAY CHSTATUS den registrierten Namen des Zertifikatsinhabers des fernen Clientzertifikats an.

Von Zertifizierungsstelle signierte Zertifikate für die gegenseitige Authentifizierung von Client und Warteschlangenmanager verwenden

Folgen Sie diesen Beispielanweisungen, um die gegenseitige Authentifizierung zwischen einem Client und einem Warteschlangenmanager mithilfe von SSL- oder TLS-Zertifikaten einer Zertifizierungsstelle zu implementieren.

Informationen zu diesem Vorgang

Szenario:

- Sie haben Client C1 und Warteschlangenmanager QM1, die sicher miteinander kommunizieren müssen. Deswegen sollen sich C1 und QM1 gegenseitig authentifizieren.
- Dieses Netz wollen Sie später in einer Produktionsumgebung verwenden, weshalb Sie von Anfang an CA-signierte Zertifikate verwenden möchten.

Die Konfiguration wird danach so aussehen:

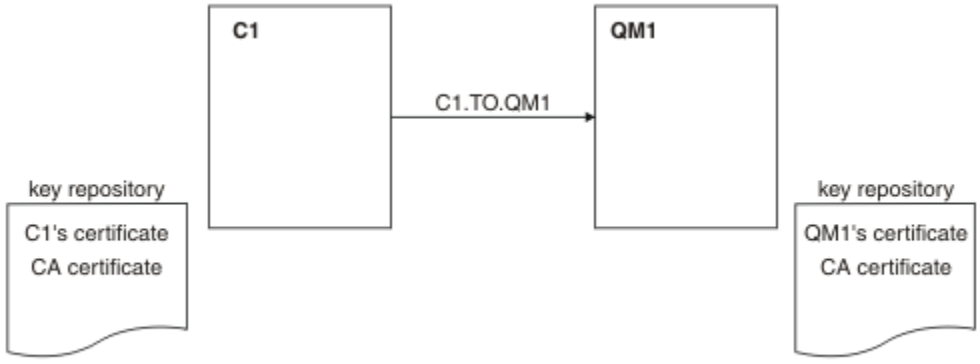


Abbildung 18. Aus dieser Task entstehende Konfiguration

In Abbildung 18 auf Seite 227 enthält das Schlüsselrepository von C1 das Zertifikat für C1 sowie das Zertifikat der Zertifizierungsstelle. Das Schlüsselrepository von QM1 enthält das Zertifikat für QM1 und das Zertifikat der Zertifizierungsstelle. In diesem Beispiel wurden die Zertifikate von C1 und QM1 von der gleichen Zertifizierungsstelle ausgestellt. Wenn die beiden CA-Zertifikate von zwei verschiedenen Zertifizierungsstellen ausgestellt worden wären, müssten die Schlüsselrepositorys von C1 und QM1 beide Zertifikate enthalten.

Vorgehensweise

1. Bereiten Sie das Schlüsselrepository auf dem Client und dem Warteschlangenmanager entsprechend des Betriebssystems vor:
   - Auf UNIX-, Linux-und Windows -Systemen.
2. Fordern Sie für den Client und den Warteschlangenmanager ein von einer Zertifizierungsstelle signiertes Zertifikat an.

   Sie können die Zertifikate von zwei verschiedenen Zertifizierungsstellen anfordern.
   - Auf UNIX-, Linux-und Windows -Systemen.
3. Fügen Sie auf dem Client und dem Warteschlangenmanager jeweils das CA-Zertifikat zum Schlüsselrepository hinzu.

   Wenn Sie für den Client und den Warteschlangenmanager verschiedene Zertifizierungsstellen verwenden, müssen Sie das CA-Zertifikat jeder Zertifizierungsstelle beiden Schlüsselrepositorys hinzufügen.
   - Auf UNIX-, Linux-und Windows -Systemen.
4. Fügen Sie auf dem Client und dem Warteschlangenmanager das von der Zertifizierungsstelle signierte Zertifikat zum Schlüsselrepository hinzu:
   - Auf UNIX-, Linux-und Windows -Systemen.
5. Definieren Sie mit einer der folgenden Methoden einen Clientverbindungskanal:
   - Verwendung des MQCONNX-Aufrufs mit der MQSCO-Struktur auf C1, wie unter Clientverbindungskanal auf dem WebSphere MQ MQI-Client erstellenbeschrieben.
   - Verwendung einer Definitionstabelle für Clientkanäle, wie unter Serververbindungs-und Clientverbindungsdefinitionen auf dem Server erstellen beschrieben.
6. Definieren Sie auf QM1 einen Serververbindungskanal, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   Der Kanal muss denselben Namen wie der in Schritt 6 definierte Clientverbindungskanal haben und die gleiche CipherSpec verwenden.

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden erstellt, wie in Abbildung 18 auf Seite 227 gezeigt.

Nächste Schritte

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe von DISPLAY-Befehlen. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie im folgenden Beispiel aus.

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
```

```
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

Das Feld SSLPEER in der Ausgabe DISPLAY CHSTATUS zeigt den SubjektDN des fernen Clientzertifikats an, der in Schritt 2 erstellt wurde. Der Name des Ausstellers entspricht dem DN des CA-Zertifikats, das das persönliche Zertifikat, das in Schritt 4 hinzugefügt wurde, signiert hat.

Client anonym mit einem Warteschlangenmanager verbinden

Folgen Sie diesen Beispielanweisungen, um ein System mit gegenseitiger Authentifizierung so zu ändern, dass sich Warteschlangenmanager anonym miteinander verbinden können.

Informationen zu diesem Vorgang

Szenario:

- Warteschlangenmanager (QM1) und Client (C1) sind eingerichtet, wie in „Von Zertifizierungsstelle signierte Zertifikate für die gegenseitige Authentifizierung von Client und Warteschlangenmanager verwenden” auf Seite 227 beschrieben.
- Sie möchten C1 so ändern, dass er sich anonym mit QM1 verbinden kann.

Die Konfiguration wird danach so aussehen:

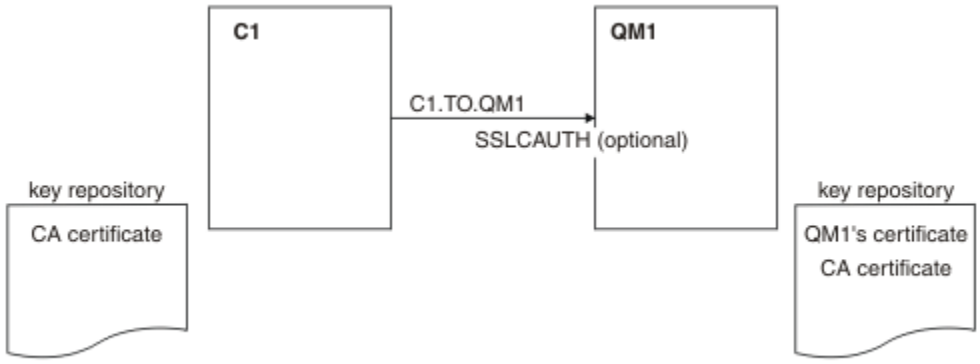


Abbildung 19. Client und Warteschlangenmanager lassen eine anonyme Verbindung zu

Vorgehensweise

1. Entfernen Sie je nach Betriebssystem das persönliche Zertifikat aus dem Schlüsselrepository für C1:
   - Auf UNIX-, Linux-und Windows -Systemen. Das Zertifikat ist wie folgt gekennzeichnet:
     - `ibmwebspheremq` gefolgt von Ihrer Anmelde-Benutzer-ID in Kleinbuchstaben, z. B. `ibmwebspheremqmyuserid`.
2. Starten Sie die Clientanwendung neu oder sorgen Sie dafür, dass die Clientanwendung geschlossen wird, und öffnen Sie dann alle SSL- oder TLS-Verbindungen neu.
3. Erlauben Sie für den Warteschlangenmanager anonyme Verbindungen, indem Sie den folgenden Befehl ausgeben:

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden geändert, wie in Abbildung 19 auf Seite 229 gezeigt.

Nächste Schritte

Das Vorhandensein des Parameterwerts für den Peer-Namen im Kanalstatus am Serverende des Kanals ist ein Hinweis darauf, dass ein Clientzertifikat übertragen wurde.

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe einiger DISPLAY-Befehle. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie im folgenden Beispiel aus:

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

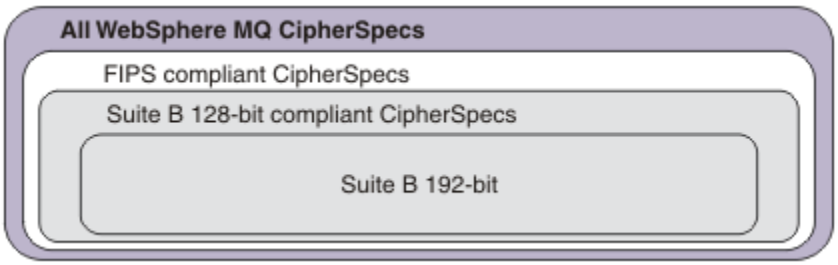
Die Felder SSLCERTI und SSLPEER sind leer, da von C1 kein Zertifikat gesendet wurde.

CipherSpecs angeben

Geben Sie eine CipherSpec an, indem Sie den Parameter **SSLCIPH** im MQSC-Befehl **DEFINE CHANNEL** oder im MQSC-Befehl **ALTER CHANNEL** verwenden.

Einige der CipherSpecs, die mit IBM WebSphere MQ verwendet werden können, sind FIPS-konform. Andere, zum Beispiel NULL_MD5, sind nicht FIPS-konform. In ähnlicher Weise sind einige der FIPS-konformen CipherSpecs auch Suite B-konform, andere jedoch nicht. 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:



In der folgenden Tabelle sind die Verschlüsselungsspezifikationen aufgeführt, die Sie mit der SSL- und TLS-Unterstützung von IBM WebSphere MQ verwenden können. Wenn Sie ein persönliches Zertifikat anfordern, geben Sie eine Schlüsselgröße für das öffentliche und das private Schlüsselpaar an. Die Schlüsselgröße, die während des SSL-Handshakes verwendet wird, entspricht der im Zertifikat hinterlegten Größe, es sei denn, die CipherSpec bestimmt sie (siehe Tabelle).

CipherSpec-Name	Verwendetes Protokoll	MAC-Algorithmus	Verschlüsselungsalgorithmus	Verschlüsselungsbits	FIPS [1]	Suite B mit 128 Bit	Suite B mit 192 Bit
NULL_MD5 [a]	SSL 3.0	MD5	--	0	Nein	Nein	Nein
NULL_SHA [a]	SSL 3.0	SHA-1	--	0	Nein	Nein	Nein

CipherSpec-Name	Verwendetes Protokoll	MAC-Algorithmus	Verschlüsselungsalgorithmus	Verschlüsselungsbits	FIPS [1]	Suite B mit 128 Bit	Suite B mit 192 Bit
RC4_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC4	40	Nein	Nein	Nein
RC4_MD5_US [a]	SSL 3.0	MD5	RC4	128	Nein	Nein	Nein
RC4_SHA_US [a]	SSL 3.0	SHA-1	RC4	128	Nein	Nein	Nein
RC2_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC2	40	Nein	Nein	Nein
DES_SHA_EXPORT [2] [a]	SSL 3.0	SHA-1	DES	56	Nein	Nein	Nein
RC4_56_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	RC4	56	Nein	Nein	Nein
DES_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	DES	56	Nein	Nein	Nein
TLS_RSA_WITH_AES_128_CBC_SHA [a]	TLS 1.0	SHA-1	AES	128	Ja	Nein	Nein
TLS_RSA_WITH_AES_256_CBC_SHA [4] [a]	TLS 1.0	SHA-1	AES	256	Ja	Nein	Nein
TLS_RSA_WITH_DES_CBC_SHA [a]	TLS 1.0	SHA-1	DES	56	Nein [5]	Nein	Nein
FIPS_WITH_DES_CBC_SHA [b]	SSL 3.0	SHA-1	DES	56	Nein [6]	Nein	Nein
TLS_RSA_WITH_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Ja	Nein	Nein
TLS_RSA_WITH_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Ja	Nein	Nein
TLS_RSA_WITH_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Ja	Nein	Nein
TLS_RSA_WITH_AES_256_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	256	Ja	Nein	Nein
ECDHE_ECDSA_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	Nein	Nein	Nein
ECDHE_RSA_RC4_128_SHA256 [b]	TLS 1.2	SHA_1	RC4	128	Nein	Nein	Nein
ECDHE_ECDSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Ja	Nein	Nein
ECDHE_ECDSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	Ja	Nein	Nein
ECDHE_RSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Ja	Nein	Nein
ECDHE_RSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	Ja	Nein	Nein
ECDHE_ECDSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Ja	Ja	Nein

CipherSpec-Name	Verwendetes Protokoll	MAC-Algorithmus	Verschlüsselungsalgorithmus	Verschlüsselungsbits	FIPS [1]	Suite B mit 128 Bit	Suite B mit 192 Bit
ECDHE_ECDSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Ja	Nein	Ja
ECDHE_RSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Ja	Nein	Nein
ECDHE_RSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Ja	Nein	Nein
TLS_RSA_WITH_NULL_SHA256 [b]	TLS 1.2	SHA-256	--	0	Nein	Nein	Nein
ECDHE_RSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	--	0	Nein	Nein	Nein
ECDHE_ECDSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	--	0	Nein	Nein	Nein
TLS_RSA_WITH_NULL_NULL [b]	TLS 1.2	--	--	0	Nein	Nein	Nein
TLS_RSA_WITH_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	Nein	Nein	Nein

Anmerkungen:

1. 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.
2. 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.
3. Die Größe des Handshakeschlüssels beträgt 1024 Bit.
4. Mithilfe dieser Verschlüsselungsspezifikation (CipherSpec) kann eine Verbindung von WebSphere MQ Explorer zu einem Warteschlangenmanager nicht geschützt werden, es sei denn, für die vom Explorer verwendete JRE gelten die entsprechenden uneingeschränkten Richtliniendateien.
5. Diese CipherSpec wurde vor dem 19. Mai 2007 FIPS 140-2-zertifiziert.
6. 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.
7. Mit dieser CipherSpec können bis zu 32 GB Daten übertragen werden, bevor die Verbindung mit Fehler AMQ9288 beendet wird. Um diesen Fehler zu vermeiden, sollten Sie Triple DES nicht verwenden oder bei Verwendung dieser CipherSpec die Rücksetzung der geheimen Schlüssel ermöglichen.

Plattformunterstützung:

- a Auf allen unterstützten Plattformen verfügbar.
- b Nur auf UNIX, Linux, and Windows-Plattformen verfügbar.

Zugehörige Konzepte

„Digitale Zertifikate und CipherSpec -Kompatibilität in IBM WebSphere MQ" auf Seite 36
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 WebSphere MQ erläutert.

Zugehörige Verweise

CHANNEL DEFINE CHANNEL

ALTER CHANNEL

Nicht weiter unterstützte CipherSpecs

Eine Liste der veralteten CipherSpecs , die Sie bei Bedarf mit WebSphere MQ verwenden können.

Weitere Informationen darüber, wie Sie veraltete CipherSpecs aktivieren können, finden Sie unter „In IBM WebSphere MQ unterstützte CipherSpec -Werte" auf Seite 41.

In der folgenden Tabelle sind die veralteten CipherSpecs aufgeführt, die Sie mit der TLS-Unterstützung von WebSphere MQ verwenden können:

Plattformunterstützung „1" auf Seite 235	CipherSpec-Name	Verwendetes Protokoll	Datenintegrität	Verschlüsselungsalgorithmus	Verschlüsselungsbits	FIPS „2" auf Seite 235	Suite B	Aktualisierung bei veralteter Version
Alle	DES_SHA_EXPORT „3" auf Seite 235	SSL 3.0	SHA-1	DES	56	Nein	Nein	7.5.0.6
Windows UNIX Linux	DES_SHA_EXPORT1024 „4" auf Seite 235	SSL 3.0	SHA-1	DES	56	Nein	Nein	7.5.0.6
Windows UNIX Linux	FIPS_WITH_DES_CBC_SHA	SSL 3.0	SHA-1	DES	56	Nein „6" auf Seite 235	Nein	7.5.0.6
Windows UNIX Linux	FIPS_WITH_3DES_EDE_CBC_SHA	SSL 3.0	SHA-1	3DES	168	Nein „7" auf Seite 235	Nein	7.5.0.8
Alle	NULL_MD5	SSL 3.0	MD5	--	0	Nein	Nein	7.5.0.6
Alle	NULL_SHA	SSL 3.0	SHA-1	--	0	Nein	Nein	7.5.0.6
Alle	RC2_MD5_EXPORT „3" auf Seite 235	SSL 3.0	MD5	RC2	40	Nein	Nein	7.5.0.7
Alle	RC4_MD5_EXPORT „3" auf Seite 235	SSL 3.0	MD5	RC4	40	Nein	Nein	7.5.0.7
Alle	RC4_MD5_US	SSL 3.0	MD5	RC4	128	Nein	Nein	7.5.0.7
Alle	RC4_SHA_US	SSL 3.0	SHA-1	RC4	128	Nein	Nein	7.5.0.7
Windows UNIX Linux	RC4_56_SHA_EXPORT1024 „4" auf Seite 235	SSL 3.0	SHA-1	RC4	56	Nein	Nein	7.5.0.7

Plattformunterstützung „1" auf Seite 235	CipherSpec-Name	Verwendetes Protokoll	Dateningtegrität	Verschlüsselungsalgorithmus	Verschlüsselungsbits	FIPS „2" auf Seite 235	Suite B	Aktualisierung bei veralteter Version
Alle	TRIPLE_DES_SHA_US	SSL 3.0	SHA-1	3DES	168	Nein	Nein	7.5.0.8
Alle	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	Nein „5" auf Seite 235	Nein	7.5.0.6
Windows UNIX Linux	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-1	--	0	Nein	Nein	7.5.0.6
Windows UNIX Linux	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	Nein	Nein	7.5.0.7
Windows UNIX Linux	ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-1	--	0	Nein	Nein	7.5.0.6
Windows UNIX Linux	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	Nein	Nein	7.5.0.7
Windows UNIX Linux	TLS_RSA_WITH_NULL_NULL	TLS 1.2	--	--	0	Nein	Nein	7.5.0.6
Alle	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	--	0	Nein	Nein	7.5.0.6
Windows UNIX Linux	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-1	RC4	128	Nein	Nein	7.5.0.7
Alle	TLS_RSA_WITH_3DES_EDE_CBC_SHA „8" auf Seite 235	TLS 1.0	SHA-1	3DES	168	Ja	Nein	7.5.0.8
Windows UNIX Linux	ECDHE_ECDSA_3DES_EDE_CBC_SHA256 „8" auf Seite 235	TLS 1.2	SHA-1	3DES	168	Ja	Nein	7.5.0.8
Windows UNIX Linux	ECDHE_RSA_3DES_EDE_CBC_SHA256 „8" auf Seite 235	TLS 1.2	SHA-1	3DES	168	Ja	Nein	7.5.0.8

Plattformunterstützung „1” auf Seite 235	CipherSpec-Name	Verwendetes Protokoll	Dateninteg rität	Verschlüsselungsalgorithmus	Verschlüsselungsbits	FIPS „2” auf Seite 235	Suite B	Aktualisierung bei veralteter Version

Anmerkungen:

1. Ist keine bestimmte Plattform angegeben, ist die CipherSpec auf allen Plattformen verfügbar.
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. 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.
4. Die Größe des Handshakeschlüssels beträgt 1024 Bit.
5. Diese CipherSpec wurde vor dem 19. Mai 2007 FIPS 140-2-zertifiziert.
6. 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.
7. 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.
8. Mit dieser CipherSpec können bis zu 32 GB Daten übertragen werden, bevor die Verbindung mit Fehler AMQ9288 beendet wird. Um diesen Fehler zu vermeiden, sollten Sie Triple DES nicht verwenden oder bei Verwendung dieser CipherSpec die Rücksetzung der geheimen Schlüssel ermöglichen.

Informationen zu CipherSpecs mit IBM WebSphere MQ Explorer abrufen

Sie können IBM WebSphere MQ Explorer verwenden, um Beschreibungen von CipherSpecs anzuzeigen.

Gehen Sie wie folgt vor, um Informationen zu den CipherSpecs in „CipherSpecs angeben” auf Seite 230 abzurufen:

1. Öffnen Sie den **IBM WebSphere 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.

Alternativen für die Angabe von CipherSpecs

Für Plattformen, auf denen SSL-Unterstützung vom Betriebssystem bereitgestellt wird, werden eventuell auch neue CipherSpecs unterstützt. Sie können eine neue CipherSpec mit dem Parameter SSLCIPH angeben, aber der von Ihnen angegebene Wert hängt von Ihrer Plattform ab.

Anmerkung: Dieser Abschnitt gilt nicht für UNIX-, Linux -oder Windows -Systeme, da die CipherSpecs mit dem Produkt WebSphere MQ bereitgestellt werden, sodass neue CipherSpecs nach dem Versand nicht verfügbar sind.

Auf Plattformen, auf denen die SSL-Unterstützung vom Betriebssystem zur Verfügung gestellt wird, werden eventuell auch neue CipherSpecs unterstützt, die nicht in „CipherSpecs angeben” auf Seite 230 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 Angabe einer SSL-CipherSpec entsprechen, die für die SSL-Version auf Ihrem System gültig ist und von ihr unterstützt wird.

IBM i
Eine Zeichenfolge mit zwei Zeichen, die einen Hexadezimalwert darstellt.

Weitere Informationen zu den zulässigen Werten finden Sie in der entsprechenden Produktdokumentation (suchen Sie nach *cipher_spec* in der IBM i -Produktdokumentation).

Sie können entweder den Befehl CHGMQMCHL oder den Befehl CRTMQMCHL verwenden, um den Wert anzugeben, z. B.:

```
CRTMQMCHL CHLNAME('channel name') SSLCIPH('hexadecimal value')
```

Sie können auch den MQSC-Befehl ALTER QMGR verwenden, um den Parameter SSLCIPH festzulegen.

z/OS
Eine Zeichenfolge mit zwei Zeichen, die einen Hexadezimalwert darstellt. Die hexadezimalen Codes entsprechen den im SSL-Protokoll definierten Werten.

Weitere Informationen finden Sie in der Beschreibung von gsk_environment_open () im API-Referenzkapitel von *z/OS Cryptographic Services System SSL Programming*, SC24-5901. Dort finden Sie eine Liste aller unterstützten SSL V3.0 -und TLS V1.0 -Verschlüsselungsspezifikationen in Form von zweistelligen hexadezimalen Codes.

Hinweise zu WebSphere MQ -Clustern

Bei WebSphere MQ -Clustern ist es am sichersten, die CipherSpec -Namen in „CipherSpecs angeben” auf Seite 230zu verwenden. 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 und Cluster” auf Seite 263.

CipherSpec für einen IBM WebSphere MQ MQI-Client angeben

Sie haben drei Optionen zur Angabe einer CipherSpec für einen IBM WebSphere 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.
- Active Directory verwenden (auf Windows -Systemen mit Active Directory -Unterstützung)

CipherSuite mit IBM WebSphere MQ -Klassen für Java und IBM WebSphere MQ -Klassen für JMS angeben

IBM WebSphere MQ -Klassen für Java und IBM WebSphere MQ -Klassen für JMS geben CipherSuites anders an als andere Plattformen.

Informationen zur Angabe einer CipherSuite mit IBM WebSphere MQ -Klassen für Java finden Sie im Abschnitt Unterstützung für Secure Sockets Layer (SSL).

Informationen zur Angabe einer CipherSuite mit IBM WebSphere MQ Classes for JMS finden Sie im Abschnitt Secure Sockets Layer (SSL) mit WebSphere MQ Classes for JMS.

Zurücksetzen von geheimen SSL-und TLS-Schlüsseln

IBM WebSphere MQ unterstützt das Zurücksetzen von geheimen Schlüsseln für Warteschlangenmanager und Clients..

Geheime Schlüssel werden zurückgesetzt, wenn eine vorgegebene Anzahl verschlüsselter Datenbyte über den Kanal geflossen ist oder nachdem der Kanal eine Zeit lang inaktiv war.

Der Wert für das Zurücksetzen des Schlüssels wird immer von der Initialisierungsseite des 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.

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 unverschlüsselten Byte darstellt, welche innerhalb eines SSL- oder TLS-Datenaustauschs gesendet und empfangen werden, bevor der geheime SSL- oder TLS-Schlüssel neu ausgehandelt wird. Der Wert 0 gibt an, dass geheime SSL- oder TLS-Schlüssel nie neu ausgehandelt werden. Wenn Sie für die Anzahl der Rücksetzungen von geheimen SSL- oder TLS-Schlüsseln einen Wert im Bereich von 1 Byte bis 32 KB setzen, verwenden die SSL- bzw. TLS-Kanäle als Zählerstand für die Rücksetzung des geheimen Schlüssels 32 KB. Dadurch werden übermäßig viele Schlüsselzurücksetzungen vermieden, zu denen es bei niedrigen Rücksetzwerten für geheime SSL- oder TLS-Schlüssel kommen würde.

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 SSL oder TLS in der Clientkonfigurationsdatei finden Sie unter SSL-Zeilengruppe der Clientkonfigurationsdatei.

Java

Für IBM WebSphere 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 sslResetoder der Umgebungseigenschaft MQC.SSL_RESET_COUNT_PROPERTY stellt die Gesamtzahl der Bytes dar, die vom WebSphere MQ Classes for Java-Clientcode gesendet und empfangen wurden, bevor der geheime Schlüssel neu 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 vom WebSphere MQ -Client gesendet und empfangen werden.

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 WebSphere MQ classes for JMS stellt die Eigenschaft "SSLRESETCOUNT" die Gesamtzahl der Byte dar, die von einer Verbindung gesendet und empfangen werden, bevor der geheime Schlüssel, der für die Verschlüsselung verwendet wird, neu 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 enthält auch Steuerinformationen, die von IBM WebSphere MQ classes for JMS gesendet und empfangen wurden. Beispiel: Um ein Verbindungsfactory-Objekt zu konfigurieren, das zum Herstellen einer Verbindung über einen für SSL oder TLS aktivierten MQI-Kanal mit einem geheimen Schlüssel verwendet werden kann, der nach 4 MB gesendeten und empfangenen Daten erneut vereinbart wird, müssen Sie den folgenden Befehl an JMSAdmin ausgeben:

```
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 gibt die ganzzahlige Eigenschaft SSLKeyResetdie Anzahl der unverschlüsselten Byte an, die innerhalb eines SSL-oder TLS-Dialogs gesendet und empfangen werden, bevor der geheime Schlüssel neu vereinbart wird.

Informationen zur Verwendung von Objekteigenschaften in IBM WebSphere MQ Classes for .NET finden Sie unter Attributwerte abrufen und festlegen.

XMS .NET

Informationen zu nicht verwalteten XMS .NET-Clients finden Sie unter Sichere Verbindungen zu einem IBM WebSphere MQ -Warteschlangenmanager.

Zugehörige Verweise
ALTER QMGR
ANZEIGEN QMGR

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-Sicher-

heitsexit 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 238.

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.

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 SSL oder TLS verwenden, bestimmt Ihre CipherSpec-Auswahl die Ebene der Datenintegrität im Unternehmen. Bei Verwendung von WebSphere MQ Advanced Message Service (AMS) können Sie die Integrität einer eindeutigen 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 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 Wert ist,

Zwei Warteschlangenmanager über SSL oder TLS verbinden

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. Zudem müssen Sie Ihre digitalen Zertifikate anfordern und verwalten. Auf einem Testsystem können Sie selbst signierte Zertifikate verwenden, die von einer lokalen Zertifizierungsstelle ausgegeben wurden. Verwenden Sie selbst signierte Zertifikate nicht auf einem Produktionssystem. Weitere Informationen finden Sie unter ../zs14140_.dita.

Vollständige Informationen zum Erstellen und Verwalten von Zertifikaten finden Sie in „Mit SSL oder TLS auf UNIX, Linux, and Windows -Systemen arbeiten" auf Seite 120.

In diesen Themen werden die Aufgaben bei der Einrichtung der SSL-Kommunikation erläutert, zudem erhalten Sie dort eine schrittweise Anleitung zum Ausführen dieser Aufgaben.

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 WebSphere MQ fordert der SSL-oder TLS-Server immer ein Zertifikat vom Client an.

Anmerkungen:

1. In diesem Kontext bezieht sich ein SSL-Client auf die Verbindung, die den Handshake initialisiert.
2. Weitere Details finden Sie im Glossar .

Auf UNIX-, Linux -und Windows -Systemen sendet der SSL-oder TLS-Client nur dann ein Zertifikat, wenn es über ein Zertifikat im richtigen WebSphere MQ -Format verfügt, das `ibmwebspheremq` gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben lautet. Beispiel: `ibmwebspheremqqm1` für `QM1`.

WebSphere MQ verwendet das Präfix `ibmwebspheremq` in einem Kennsatz, um Verwechslungen 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. Sendet der Client kein Zertifikat, schlägt die Authentifizierung nur dann fehl, wenn der Parameter SSLCAUTH für das Ende des Kanals, das als SSL- oder TLS-Server agiert, auf REQUIRED gesetzt ist oder ein Wert für den Parameter SSLPEER angegeben ist. Weitere Informationen zur anonymen Verbindung eines Warteschlangenmanagers (d. h. wenn der SSL- oder TLS-Client kein Zertifikat sendet) finden Sie im Abschnitt „Zwei Warteschlangenmanager über unidirektionale Authentifizierung verbinden" auf Seite 223.

Digitale Zertifikatsetiketten, Kenntnisse der Anforderungen

Wenn Sie SSL und TLS für die Verwendung digitaler Zertifikate einrichten, gibt es möglicherweise bestimmte Bezeichnungsanforderungen, die je nach verwendeter Plattform und Verbindungsmethode erfüllt werden müssen.

Informationen zu diesem Vorgang

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 die Felder *Subject Distinguished Name* und *Subject Common Name* im Zertifikat selbst enthalten sind. Diese werden definiert, wenn das Zertifikat erstellt wird und nicht geändert werden kann. Sie können jedoch bei Bedarf die Bezeichnung ändern, die einem digitalen Zertifikat zugeordnet ist.

Wie wird die Zertifikatsbezeichnung verwendet?

IBM WebSphere MQ verwendet Zertifikatsbezeichnungen, um ein persönliches Zertifikat zu suchen, das während des SSL-Handshakes gesendet wird. Dies eliminiert Mehrdeutigkeiten, wenn mehr als ein persönliches Zertifikat im Schlüsselrepository vorhanden ist.

Zertifikatsbezeichnungen folgen einer Namenskonvention. Sie müssen sicherstellen, dass Sie die korrekte Namenskonvention für Bezeichnungen verwenden, die der verwendeten Plattform entspricht.

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

Während des SSL-oder TLS-Handshakes ruft der SSL-oder TLS-Client immer ein digitales Zertifikat vom Server ab und validiert es. Bei der IBM WebSphere MQ -Implementierung fordert der SSL-oder TLS-Server immer ein Zertifikat vom Client an und der Client stellt dem Server immer ein Zertifikat bereit, 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 SSL-oder 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, das als SSL-oder TLS-Server fungiert, mit dem auf REQUIRED gesetzten Parameter SSLCAUTH oder einem auf SSLPEER gesetzten Parameterwert definiert ist.

Weitere Informationen zur Verbindung eines Warteschlangenmanagers mit unidirektionaler Authentifizierung, d. h. wenn der SSL-oder TLS-Client kein Zertifikat sendet, finden Sie unter „Zwei Warteschlangenmanager über unidirektionale Authentifizierung verbinden” auf Seite 223.

, UNIX, Linux, and Windows -Systeme

Informationen zu diesem Vorgang

Auf -, UNIX, Linux, and Windows -Systemen sendet der SSL-oder TLS-Server ein Zertifikat nur dann an den Client, wenn der Server ein Zertifikat im richtigen IBM WebSphere MQ -Format findet. Auf diesen Systemen lautet das korrekte Format `ibmwebspheremq`, gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben.

Für einen Warteschlangenmanager mit dem Namen `QM1`lautet die Zertifikatskennsatzanforderung beispielsweise wie folgt:

```
ibmwebspheremqqm1
```

Wenn im Schlüsselrepository des Warteschlangenmanagers kein Zertifikat gefunden wird, das mit der erforderlichen Bezeichnung im richtigen Fall und Format übereinstimmt, tritt ein Fehler auf und der SSL-oder TLS-Handshake schlägt fehl.

Die IBM WebSphere MQ-Client-Datenkonvertierung

Informationen zu diesem Vorgang

Beim Herstellen einer Verbindung von einer IBM WebSphere MQ -Clientanwendung aus sendet der SSL-oder TLS-Client nur dann ein Zertifikat, wenn es ein Zertifikat mit einer Bezeichnung im Format `ibmwebspheremq`hat, gefolgt vom Benutzernamen des Benutzers, der den Clientanwendungsprozess ausführt.

Für den Benutzernamen `wasadmin`lautet die Anforderung für die Zertifikatsbezeichnung beispielsweise wie folgt in Kleinbuchstaben:

```
  ibmwebspheremqwasadmin
```

Die obige Bezeichnungsanforderung gilt für Message Service Clients for C oder C++ und .NET.

IBM WebSphere MQ Java-oder IBM WebSphere MQ JMS-Client

Informationen zu diesem Vorgang

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

Das Standardverhalten besteht darin, dass der JSSE-Client die Zertifikate im Schlüsselrepository durchläuft und das erste zulässige persönliche Zertifikat auswählt. 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 oder zum besseren Verständnis des Handshakes, der von der IBM WebSphere MQ -Java-Clientanwendung in Kombination mit Ihrem speziellen JSSE-Provider ausgeführt wird, können Sie das Debugging aktivieren, indem Sie Folgendes festlegen:

```
javax.net.debug=ssl
```

in der JVM-Umgebung.

Sie können `-Djavax.net.debug=ssl` in der Befehlszeile verwenden oder die Variable in der Anwendung oder über die Konfiguration festlegen.

Zugehörige Konzepte

„Persönliches Zertifikat in ein Schlüsselrepository auf UNIX, Linux, and Windows -Systemen importieren” auf Seite 141
Gehen Sie wie folgt vor, um ein persönliches Zertifikat zu importieren:

Selbst signierte Zertifikate für die gegenseitige Authentifizierung zweier Warteschlangenmanager verwenden

Folgen Sie diesen Beispielanweisungen, um die gegenseitige Authentifizierung zweier Warteschlangenmanager mithilfe von selbst signierten SSL- oder TLS-Zertifikaten zu implementieren.

Informationen zu diesem Vorgang

Szenario:

- Sie haben zwei Warteschlangenmanager, QM1 und QM2, die sicher miteinander kommunizieren müssen. Deswegen sollen sich QM1 und QM2 gegenseitig authentifizieren.
- Die sichere Kommunikation möchten Sie mit selbst signierten Zertifikaten testen.

Die Konfiguration wird danach so aussehen:

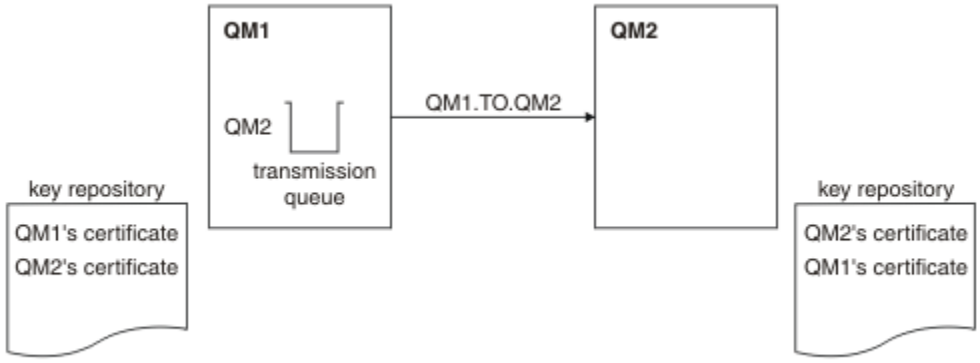


Abbildung 20. Aus dieser Task entstehende Konfiguration

In Abbildung 14 auf Seite 219 enthält das Schlüsselrepository von QM1 das Zertifikat für QM1 und das öffentliche Zertifikat von QM2. Das Schlüsselrepository von QM2 enthält das Zertifikat für QM2 und das öffentliche Zertifikat von QM1.

Vorgehensweise

1. Bereiten Sie das Schlüsselrepository auf beiden Warteschlangenmanagern entsprechend des Betriebssystems vor:
   - Auf UNIX-, Linux-und Windows -Systemen.
2. Erstellen Sie für jeden Warteschlangenmanager ein selbst signiertes Zertifikat:
   - Auf UNIX-, Linux-und Windows -Systemen.
3. Extrahieren Sie eine Kopie jedes Zertifikats:
   - Auf UNIX-, Linux-und Windows -Systemen.
4. Übertragen Sie den öffentlichen Teil des Zertifikats QM1 mit einem Dienstprogramm wie FTP).
5. Fügen Sie auf jedem Warteschlangenmanager das Partnerzertifikat zum Schlüsselrepository hinzu:
   - Auf UNIX-, Linux-und Windows -Systemen.
6. Definieren Sie auf QM1 einen Senderkanal und die zugehörige Übertragungswarteschlange, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
   SSLCIPH(RC4_MD5_US) DESCR('Sender channel using SSL from QM1 to QM2')

   DEFINE QLOCAL(QM2) USAGE(XMITQ)
   ```

   Dieses Beispiel verwendet die CipherSpec RC4_MD5. Die CipherSpecs an jedem Ende des Kanals müssen identisch sein.
7. Definieren Sie auf QM2 einen Empfängerkanal, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from QM1 to QM2')
   ```

   Der Kanal muss denselben Namen wie der in Schritt 6 definierte Senderkanal haben und die gleiche CipherSpec verwenden.
8. Starten Sie den Kanalbeschrieben.

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden erstellt, wie in Abbildung 14 auf Seite 219 gezeigt.

Nächste Schritte

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe von DISPLAY-Befehlen. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie in den folgenden Beispielen aus.

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM1.TO.QM2)                     CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

Geben Sie auf dem Warteschlangenmanager QM2 den folgenden Befehl ein:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(QM2.TO.QM1)                     CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QM1)
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

In jedem Fall muss der Wert von SSLPEER mit dem DN im Partnerzertifikat übereinstimmen, das in Schritt 2 erstellt wurde. Da es sich um ein selbst signiertes Zertifikat handelt, entspricht der Name des Ausstellers dem Namen der Partnerwarteschlange.

SSLPEER ist optional. Wenn dieses Attribut allerdings angegeben ist, muss sein Wert den DN des in Schritt 2 erstellten Partnerzertifikat zulassen. Weitere Informationen zur Verwendung von SSLPEER finden Sie unter WebSphere MQ -Regeln für SSLPEER-Werte .

Von Zertifizierungsstelle signierte Zertifikate für die gegenseitige Authentifizierung zweier Warteschlangenmanager verwenden

Folgen Sie diesen Beispielanweisungen, um die gegenseitige Authentifizierung zweier Warteschlangenmanager mithilfe von SSL- oder TLS-Zertifikaten einer Zertifizierungsstelle zu implementieren.

Informationen zu diesem Vorgang

Szenario:

- Sie haben zwei Warteschlangenmanager, QMA und QMB, die sicher miteinander kommunizieren müssen. Deswegen sollen sich QMA und QMB gegenseitig authentifizieren.
- Dieses Netz wollen Sie später in einer Produktionsumgebung verwenden, weshalb Sie von Anfang an CA-signierte Zertifikate verwenden möchten.

Die Konfiguration wird danach so aussehen:

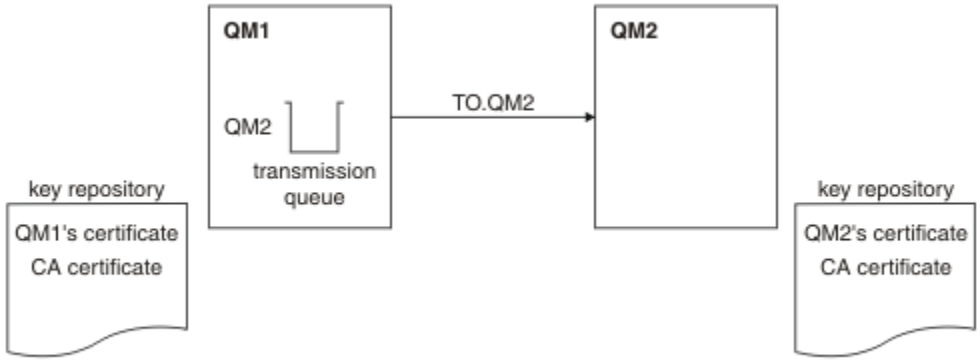


Abbildung 21. Aus dieser Task entstehende Konfiguration

In Abbildung 15 auf Seite 221 enthält das Schlüsselrepository von QMA das Zertifikat für QMA sowie das Zertifikat der Zertifizierungsstelle. Das Schlüsselrepository von QMB enthält das Zertifikat für QMB und das Zertifikat der Zertifizierungsstelle. In diesem Beispiel wurden die Zertifikate von QMA und QMB von der gleichen Zertifizierungsstelle ausgestellt. Wenn die beiden CA-Zertifikate von zwei verschiedenen Zertifizierungsstellen ausgestellt worden wären, müssten die Schlüsselrepositorys von QMA und QMB beide Zertifikate enthalten.

Vorgehensweise

1. Bereiten Sie das Schlüsselrepository auf beiden Warteschlangenmanagern entsprechend des Betriebssystems vor:
   - Auf UNIX-, Linux-und Windows -Systemen.
2. Fordern Sie für jeden Warteschlangenmanager ein von einer Zertifizierungsstelle signiertes Zertifikat an.

   Sie können die Zertifikate von zwei verschiedenen Zertifizierungsstellen anfordern.
   - Auf UNIX-, Linux-und Windows -Systemen.
3. Fügen Sie das CA-Zertifikat zum Schlüsselrepository des jeweiligen Warteschlangenmanagers hinzu:

   Wenn Sie für die Warteschlangenmanager verschiedene Zertifizierungsstellen verwenden, müssen Sie das CA-Zertifikat jeder Zertifizierungsstelle beiden Schlüsselrepositorys hinzufügen.
   - Auf UNIX-, Linux-und Windows -Systemen.
4. Fügen Sie auf jedem Warteschlangenmanager das von der Zertifizierungsstelle signierte Zertifikat zum Schlüsselrepository hinzu:
   - Auf UNIX-, Linux-und Windows -Systemen.
5. Definieren Sie auf QMA einen Senderkanal und die zugehörige Übertragungswarteschlange, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
   SSLCIPH(RC2_MD5_EXPORT) DESCR('Sender channel using SSL from QMA to QMB')

   DEFINE QLOCAL(QMB) USAGE(XMITQ)
   ```

   Dieses Beispiel verwendet die CipherSpec RC4_MD5. Die CipherSpecs an jedem Ende des Kanals müssen identisch sein.
6. Definieren Sie auf QMB einen Empfängerkanal, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL to QMB')
   ```

   Der Kanal muss denselben Namen wie der in Schritt 6 definierte Senderkanal haben und die gleiche CipherSpec verwenden.

7. Starten Sie den Kanal:

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden erstellt, wie in Abbildung 15 auf Seite 221 gezeigt.

Nächste Schritte

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe von DISPLAY-Befehlen. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie in den folgenden Beispielen aus.

Geben Sie auf dem Warteschlangenmanager QMA den folgenden Befehl ein:

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QMB)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QMB)
```

Geben Sie auf dem Warteschlangenmanager QMB den folgenden Befehl ein:

```
DISPLAY CHS(TO.QMB) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QMB)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
   XMITQ( )
```

In jedem Fall muss der Wert von SSLPEER mit dem Wert des definierten Namens (DN) im Partnerzertifikat übereinstimmen, der in Schritt 2 erstellt wurde. Der Name des Ausstellers entspricht dem DN des CA-Zertifikats, das das persönliche Zertifikat, das in Schritt 4 hinzugefügt wurde, signiert hat.

Zwei Warteschlangenmanager über unidirektionale Authentifizierung verbinden

Folgen Sie diesen Beispielanweisungen, um ein System mit gegenseitiger Authentifizierung so zu ändern, dass sich Warteschlangenmanager über die unidirektionale Authentifizierung miteinander verbinden können (für den Fall, dass der SSL- oder TLS-Client kein Zertifikat sendet).

Informationen zu diesem Vorgang

Szenario:

- Ihre beiden Warteschlangenmanager (QM1 und QM2) sind eingerichtet, wie in „Von Zertifizierungsstelle signierte Zertifikate für die gegenseitige Authentifizierung zweier Warteschlangenmanager verwenden" auf Seite 221 beschrieben.

- Sie möchten QM1 so ändern, dass er sich über die unidirektionale Authentifizierung mit QM2 verbinden kann.

Die Konfiguration wird danach so aussehen:

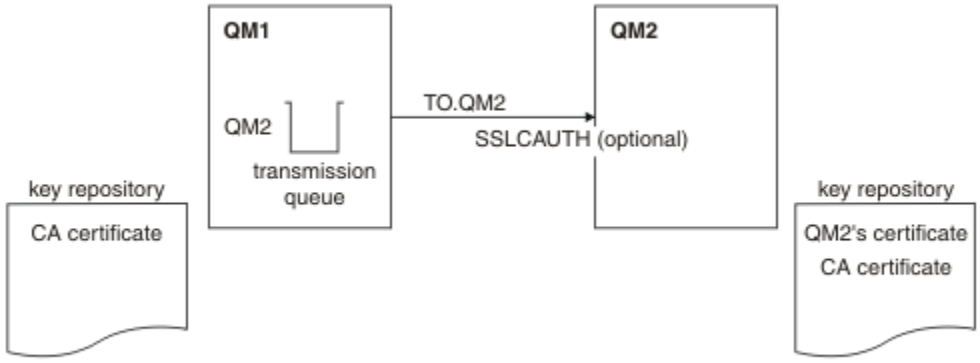


Abbildung 22. Warteschlangenmanager lassen unidirektionale Authentifizierung zu

Vorgehensweise

1. Entfernen Sie je nach Betriebssystem das persönliche Zertifikat von QM1aus dem zugehörigen Schlüsselrepository:
   - Auf UNIX-, Linux-und Windows -Systemen. Das Zertifikat ist wie folgt gekennzeichnet:
     - `ibmwebspheremq` gefolgt vom Namen Ihres Warteschlangenmanagers in Kleinbuchstaben. Beispiel für `QM1` : `ibmwebspheremqqm1`.
2. Optional: Aktualisieren Sie auf QM1, wenn zuvor SSL-oder TLS-Kanäle ausgeführt wurden, die SSL- oder TLS-Umgebungbeschrieben.
3. Erlauben Sie anonyme Verbindungen auf dem Empfängerbeschrieben.

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden geändert, wie in Abbildung 16 auf Seite 223 gezeigt.

Nächste Schritte

Wenn der Senderkanal aktiv war und Sie in Schritt 2 den Befehl REFRESH SECURITY TYPE(SSL) ausgegeben haben, wird der Kanal automatisch gestartet. Wenn der Senderkanal inaktiv war, starten Sie ihn.

Das Vorhandensein des Parameterwerts für den Peer-Namen im Kanalstatus am Serverende des Kanals ist ein Hinweis darauf, dass ein Clientzertifikat übertragen wurde.

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe einiger DISPLAY-Befehle. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie in den folgenden Beispielen aus:

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
     4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(SDR)
   CONNAME(9.20.25.40)                     CURRENT
   RQMNAME(QM2)
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
```

```
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(MQGET)
   XMITQ(QM2)
```

Geben Sie auf dem Warteschlangenmanager QM2 den folgenden Befehl ein:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(TO.QM2)                         CHLTYPE(RCVR)
   CONNAME(9.20.35.92)                     CURRENT
   RQMNAME(QMA)                            SSLCERTI( )
   SSLPEER( )                              STATUS(RUNNING)
   SUBSTATE(RECEIVE)                       XMITQ( )
```

Auf QM2 ist das Feld SSLPEER leer, da von QM1 kein Zertifikat gesendet wurde. Auf QM1 stimmt der Wert von SSLPEER mit dem DN im persönlichen Zertifikat von QM2 überein.

Client sicher mit einem WS-Manager verbinden

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. Zudem müssen Sie Ihre digitalen Zertifikate anfordern und verwalten. Auf einem Testsystem können Sie selbst signierte Zertifikate verwenden, die von einer lokalen Zertifizierungsstelle ausgegeben wurden. Verwenden Sie selbst signierte Zertifikate nicht auf einem Produktionssystem. Weitere Informationen finden Sie unter ../zs14140_.dita.

Vollständige Informationen zum Erstellen und Verwalten von Zertifikaten finden Sie in „Mit SSL oder TLS auf UNIX, Linux, and Windows -Systemen arbeiten” auf Seite 120.

In diesen Themen werden die Aufgaben bei der Einrichtung der SSL-Kommunikation erläutert, zudem erhalten Sie dort eine schrittweise Anleitung zum Ausführen dieser Aufgaben.

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 WebSphere MQ fordert der SSL-oder TLS-Server immer ein Zertifikat vom Client an.

Auf UNIX, Linux, and Windows -Systemen sendet der SSL-oder TLS-Client ein Zertifikat nur, wenn es über ein Zertifikat verfügt, das im richtigen WebSphere MQ -Format beschriftet ist. Auf dieses Format folgt `ibmwebspheremq` , auf das Ihre Anmelde-Benutzer-ID in Kleinbuchstaben geändert wurde, z. B. `ibmwebspheremqmyuserid`.

WebSphere MQ verwendet das Präfix `ibmwebspheremq` in einem Kennsatz, um Verwechslungen 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. Sendet der Client kein Zertifikat, schlägt die Authentifizierung nur dann fehl, wenn der Parameter SSLCAUTH für das Ende des Kanals, das als SSL- oder TLS-Server agiert, auf REQUIRED gesetzt ist oder ein Wert für den Parameter SSLPEER angegeben ist. Weitere Informationen zur anonymen Verbindung eines Warteschlangenmanagers finden Sie unter „Client anonym mit einem Warteschlangenmanager verbinden” auf Seite 229.

Selbst signierte Zertifikate für die gegenseitige Authentifizierung von Client und Warteschlangenmanager verwenden

Folgen Sie diesen Beispielanweisungen, um die gegenseitige Authentifizierung zwischen einem Client und einem Warteschlangenmanager mithilfe von selbst signierten SSL- oder TLS-Zertifikaten zu implementieren.

Informationen zu diesem Vorgang

Szenario:

- Sie haben Client C1 und Warteschlangenmanager QM1, die sicher miteinander kommunizieren müssen. Deswegen sollen sich C1 und QM1 gegenseitig authentifizieren.
- Die sichere Kommunikation möchten Sie mit selbst signierten Zertifikaten testen.

Unter IBM i unterstützt DCM keine selbst signierten Zertifikate, weshalb diese Task auf IBM i-Systemen nicht anwendbar ist.

Die Konfiguration wird danach so aussehen:

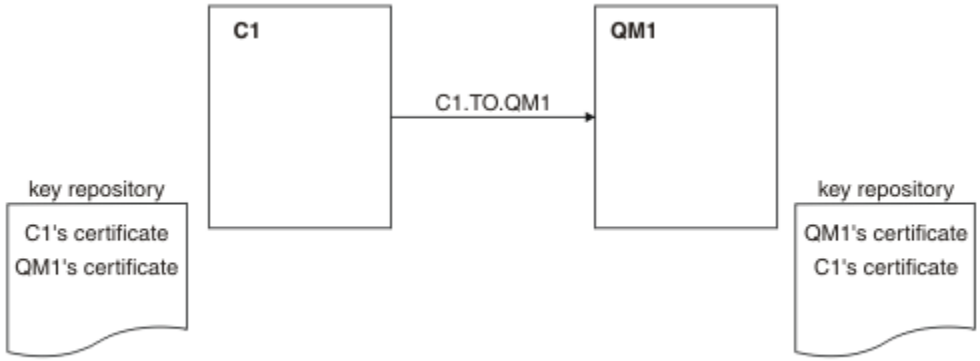


Abbildung 23. Aus dieser Task entstehende Konfiguration

In Abbildung 17 auf Seite 226 enthält das Schlüsselrepository von QM1 das Zertifikat für QM1 und das öffentliche Zertifikat von C1. Das Schlüsselrepository von C1 enthält das Zertifikat für C1 und das öffentliche Zertifikat von QM1.

Vorgehensweise

1. Bereiten Sie das Schlüsselrepository auf dem Client und dem Warteschlangenmanager entsprechend des Betriebssystems vor:
   - Auf UNIX-, Linux-und Windows -Systemen.
2. Erstellen Sie für den Client und den Warteschlangenmanager ein selbst signiertes Zertifikat:
   - Auf UNIX-, Linux-und Windows -Systemen.
3. Extrahieren Sie eine Kopie jedes Zertifikats:
   - Auf UNIX-, Linux-und Windows -Systemen.
4. Übertragen Sie den öffentlichen Teil des Zertifikats C1 auf das System QM1 und umgekehrt, indem Sie ein Dienstprogramm wie FTPbeschrieben.
5. Fügen Sie auf dem Client und dem Warteschlangenmanager jeweils das Partnerzertifikat zum Schlüsselrepository hinzu:
   - Auf UNIX-, Linux-und Windows -Systemen.
6. Geben Sie den Befehl REFRESH SECURITY TYPE(SSL) auf dem Warteschlangenmanager aus.
7. Definieren Sie mit einer der folgenden Methoden einen Clientverbindungskanal:

- Verwendung des MQCONNX-Aufrufs mit der MQSCO-Struktur auf C1, wie unter Clientverbindungskanal auf dem WebSphere MQ MQI-Client erstellenbeschrieben.
- Verwendung einer Definitionstabelle für Clientkanäle, wie unter Serververbindungs-und Clientverbindungsdefinitionen auf dem Server erstellen beschrieben.

8. Definieren Sie auf QM1 einen Serververbindungskanal, indem Sie einen Befehl wie den folgenden ausgeben:

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC4_MD5_US)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
```

   Der Kanal muss denselben Namen wie der in Schritt 6 definierte Clientverbindungskanal haben und die gleiche CipherSpec verwenden.

Ergebnisse

Schlüsselrepositorys und Kanäle werden wie in Abbildung 17 auf Seite 226 dargestellt erstellt.

Nächste Schritte

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe von DISPLAY-Befehlen. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie im folgenden Beispiel aus.

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                       CURRENT
   SSLCERTI("CN=QM1,OU=WebSphere MQ Development,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=WebSphere MQ Development,O=IBM,ST=Hamp□
shire,C=UK")
   STATUS(RUNNING)                           SUBSTATE(RECEIVE)
```

Optional können Sie für die Kanaldefinitionen auch das Filterattribut SSLPEER festlegen. Wenn die Kanaldefinition SSLPEER festgelegt ist, muss ihr Wert mit dem registrierten Namen des Zertifikatsinhabers im Partnerzertifikat übereinstimmen, das in Schritt 2 erstellt wurde. Nach einer erfolgreichen Verbindung zeigt das Feld SSLPEER in der Ausgabe von DISPLAY CHSTATUS den registrierten Namen des Zertifikatsinhabers des fernen Clientzertifikats an.

Von Zertifizierungsstelle signierte Zertifikate für die gegenseitige Authentifizierung von Client und Warteschlangenmanager verwenden

Folgen Sie diesen Beispielanweisungen, um die gegenseitige Authentifizierung zwischen einem Client und einem Warteschlangenmanager mithilfe von SSL- oder TLS-Zertifikaten einer Zertifizierungsstelle zu implementieren.

Informationen zu diesem Vorgang

Szenario:

- Sie haben Client C1 und Warteschlangenmanager QM1, die sicher miteinander kommunizieren müssen. Deswegen sollen sich C1 und QM1 gegenseitig authentifizieren.
- Dieses Netz wollen Sie später in einer Produktionsumgebung verwenden, weshalb Sie von Anfang an CA-signierte Zertifikate verwenden möchten.

Die Konfiguration wird danach so aussehen:

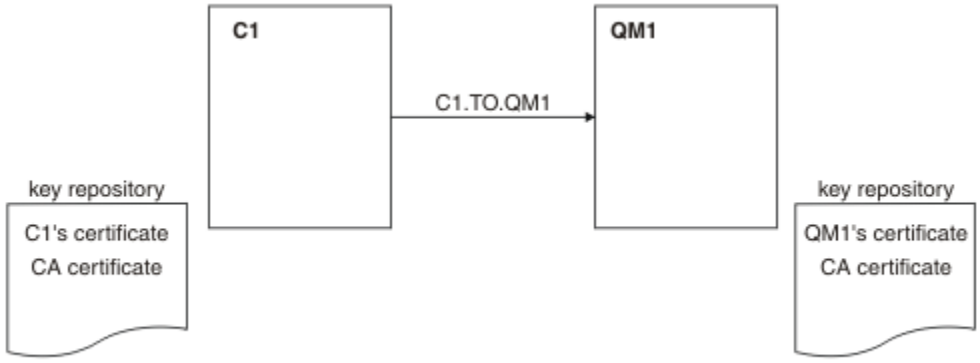


Abbildung 24. Aus dieser Task entstehende Konfiguration

In Abbildung 18 auf Seite 227 enthält das Schlüsselrepository von C1 das Zertifikat für C1 sowie das Zertifikat der Zertifizierungsstelle. Das Schlüsselrepository von QM1 enthält das Zertifikat für QM1 und das Zertifikat der Zertifizierungsstelle. In diesem Beispiel wurden die Zertifikate von C1 und QM1 von der gleichen Zertifizierungsstelle ausgestellt. Wenn die beiden CA-Zertifikate von zwei verschiedenen Zertifizierungsstellen ausgestellt worden wären, müssten die Schlüsselrepositorys von C1 und QM1 beide Zertifikate enthalten.

Vorgehensweise

1. Bereiten Sie das Schlüsselrepository auf dem Client und dem Warteschlangenmanager entsprechend des Betriebssystems vor:
   - Auf UNIX-, Linux-und Windows -Systemen.
2. Fordern Sie für den Client und den Warteschlangenmanager ein von einer Zertifizierungsstelle signiertes Zertifikat an.

   Sie können die Zertifikate von zwei verschiedenen Zertifizierungsstellen anfordern.
   - Auf UNIX-, Linux-und Windows -Systemen.
3. Fügen Sie auf dem Client und dem Warteschlangenmanager jeweils das CA-Zertifikat zum Schlüsselrepository hinzu.

   Wenn Sie für den Client und den Warteschlangenmanager verschiedene Zertifizierungsstellen verwenden, müssen Sie das CA-Zertifikat jeder Zertifizierungsstelle beiden Schlüsselrepositorys hinzufügen.
   - Auf UNIX-, Linux-und Windows -Systemen.
4. Fügen Sie auf dem Client und dem Warteschlangenmanager das von der Zertifizierungsstelle signierte Zertifikat zum Schlüsselrepository hinzu:
   - Auf UNIX-, Linux-und Windows -Systemen.
5. Definieren Sie mit einer der folgenden Methoden einen Clientverbindungskanal:
   - Verwendung des MQCONNX-Aufrufs mit der MQSCO-Struktur auf C1, wie unter Clientverbindungskanal auf dem WebSphere MQ MQI-Client erstellenbeschrieben.
   - Verwendung einer Definitionstabelle für Clientkanäle, wie unter Serververbindungs-und Clientverbindungsdefinitionen auf dem Server erstellen beschrieben.
6. Definieren Sie auf QM1 einen Serververbindungskanal, indem Sie einen Befehl wie den folgenden ausgeben:

   ```
   DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(RC2_MD5_EXPORT)
   SSLCAUTH(REQUIRED) DESCR('Receiver channel using SSL from C1 to QM1')
   ```

   Der Kanal muss denselben Namen wie der in Schritt 6 definierte Clientverbindungskanal haben und die gleiche CipherSpec verwenden.

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden erstellt, wie in Abbildung 18 auf Seite 227 gezeigt.

Nächste Schritte

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe von DISPLAY-Befehlen. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie im folgenden Beispiel aus.

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI("CN=WebSphere MQ CA,OU=WebSphere MQ Devt,O=IBM,ST=Hampshire,C=UK")
   SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=WebSphere MQ Development,O=IBM,ST=Hamp
shire,C=UK")
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

Das Feld SSLPEER in der Ausgabe DISPLAY CHSTATUS zeigt den SubjektDN des fernen Clientzertifikats an, der in Schritt 2 erstellt wurde. Der Name des Ausstellers entspricht dem DN des CA-Zertifikats, das das persönliche Zertifikat, das in Schritt 4 hinzugefügt wurde, signiert hat.

Client anonym mit einem Warteschlangenmanager verbinden

Folgen Sie diesen Beispielanweisungen, um ein System mit gegenseitiger Authentifizierung so zu ändern, dass sich Warteschlangenmanager anonym miteinander verbinden können.

Informationen zu diesem Vorgang

Szenario:

- Warteschlangenmanager (QM1) und Client (C1) sind eingerichtet, wie in „Von Zertifizierungsstelle signierte Zertifikate für die gegenseitige Authentifizierung von Client und Warteschlangenmanager verwenden” auf Seite 227 beschrieben.
- Sie möchten C1 so ändern, dass er sich anonym mit QM1 verbinden kann.

Die Konfiguration wird danach so aussehen:

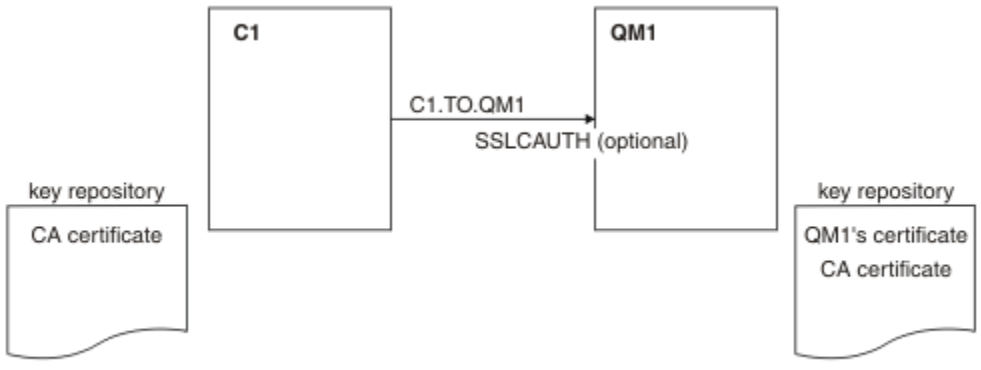


Abbildung 25. Client und Warteschlangenmanager lassen eine anonyme Verbindung zu

Vorgehensweise

1. Entfernen Sie je nach Betriebssystem das persönliche Zertifikat aus dem Schlüsselrepository für C1:
   - Auf UNIX-, Linux-und Windows -Systemen. Das Zertifikat ist wie folgt gekennzeichnet:
     – `ibmwebspheremq` gefolgt von Ihrer Anmelde-Benutzer-ID in Kleinbuchstaben, z. B. `ibmwebspheremqmyuserid`.
2. Starten Sie die Clientanwendung neu oder sorgen Sie dafür, dass die Clientanwendung geschlossen wird, und öffnen Sie dann alle SSL- oder TLS-Verbindungen neu.
3. Erlauben Sie für den Warteschlangenmanager anonyme Verbindungen, indem Sie den folgenden Befehl ausgeben:

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

Ergebnisse

Die Schlüsselrepositorys und Kanäle werden geändert, wie in Abbildung 19 auf Seite 229 gezeigt.

Nächste Schritte

Das Vorhandensein des Parameterwerts für den Peer-Namen im Kanalstatus am Serverende des Kanals ist ein Hinweis darauf, dass ein Clientzertifikat übertragen wurde.

Überprüfen Sie die erfolgreiche Ausführung der Task durch Ausgabe einiger DISPLAY-Befehle. Bei erfolgreichem Abschluss der Task sieht die Ausgabe in etwa wie im folgenden Beispiel aus:

Geben Sie auf dem Warteschlangenmanager QM1 den folgenden Befehl ein:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

Die Ausgabe sieht in etwa wie folgt aus:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
     5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
   CHANNEL(C1.TO.QM1)                      CHLTYPE(SVRCONN)
   CONNAME(9.20.35.92)                     CURRENT
   SSLCERTI( )                             SSLPEER( )
   STATUS(RUNNING)                         SUBSTATE(RECEIVE)
```

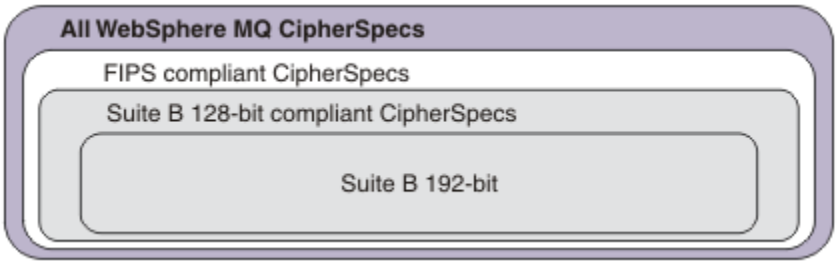
Die Felder SSLCERTI und SSLPEER sind leer, da von C1 kein Zertifikat gesendet wurde.

CipherSpecs angeben

Geben Sie eine CipherSpec an, indem Sie den Parameter **SSLCIPH** im MQSC-Befehl **DEFINE CHANNEL** oder im MQSC-Befehl **ALTER CHANNEL** verwenden.

Einige der CipherSpecs, die mit IBM WebSphere MQ verwendet werden können, sind FIPS-konform. Andere, zum Beispiel `NULL_MD5`, sind nicht FIPS-konform. In ähnlicher Weise sind einige der FIPS-konformen CipherSpecs auch Suite B-konform, andere jedoch nicht. 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:



In der folgenden Tabelle sind die Verschlüsselungsspezifikationen aufgeführt, die Sie mit der SSL- und TLS-Unterstützung von IBM WebSphere MQ verwenden können. Wenn Sie ein persönliches Zertifikat anfordern, geben Sie eine Schlüsselgröße für das öffentliche und das private Schlüsselpaar an. Die Schlüsselgröße, die während des SSL-Handshakes verwendet wird, entspricht der im Zertifikat hinterlegten Größe, es sei denn, die CipherSpec bestimmt sie (siehe Tabelle).

CipherSpec-Name	**Verwendetes Protokoll**	**MAC-Algorithmus**	**Verschlüsselungsalgorithmus**	**Verschlüsselungsbits**	**FIPS [1]**	**Suite B mit 128 Bit**	**Suite B mit 192 Bit**
NULL_MD5 [a]	SSL 3.0	MD5	--	0	Nein	Nein	Nein
NULL_SHA [a]	SSL 3.0	SHA-1	--	0	Nein	Nein	Nein
RC4_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC4	40	Nein	Nein	Nein
RC4_MD5_US [a]	SSL 3.0	MD5	RC4	128	Nein	Nein	Nein
RC4_SHA_US [a]	SSL 3.0	SHA-1	RC4	128	Nein	Nein	Nein
RC2_MD5_EXPORT [2] [a]	SSL 3.0	MD5	RC2	40	Nein	Nein	Nein
DES_SHA_EXPORT [2] [a]	SSL 3.0	SHA-1	DES	56	Nein	Nein	Nein
RC4_56_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	RC4	56	Nein	Nein	Nein
DES_SHA_EXPORT1024 [3] [b]	SSL 3.0	SHA-1	DES	56	Nein	Nein	Nein
TLS_RSA_WITH_AES_128_CBC_SHA [a]	TLS 1.0	SHA-1	AES	128	Ja	Nein	Nein
TLS_RSA_WITH_AES_256_CBC_SHA [4] [a]	TLS 1.0	SHA-1	AES	256	Ja	Nein	Nein
TLS_RSA_WITH_DES_CBC_SHA [a]	TLS 1.0	SHA-1	DES	56	Nein[5]	Nein	Nein
FIPS_WITH_DES_CBC_SHA [b]	SSL 3.0	SHA-1	DES	56	Nein[6]	Nein	Nein
TLS_RSA_WITH_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Ja	Nein	Nein

CipherSpec-Name	Verwendetes Protokoll	MAC-Algorithmus	Verschlüsselungsalgorithmus	Verschlüsselungsbits	FIPS [1]	Suite B mit 128 Bit	Suite B mit 192 Bit
TLS_RSA_WITH_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Ja	Nein	Nein
TLS_RSA_WITH_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Ja	Nein	Nein
TLS_RSA_WITH_AES_256_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	256	Ja	Nein	Nein
ECDHE_ECDSA_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	Nein	Nein	Nein
ECDHE_RSA_RC4_128_SHA256 [b]	TLS 1.2	SHA_1	RC4	128	Nein	Nein	Nein
ECDHE_ECDSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Ja	Nein	Nein
ECDHE_ECDSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	Ja	Nein	Nein
ECDHE_RSA_AES_128_CBC_SHA256 [b]	TLS 1.2	SHA-256	AES	128	Ja	Nein	Nein
ECDHE_RSA_AES_256_CBC_SHA384 [b]	TLS 1.2	SHA-384	AES	256	Ja	Nein	Nein
ECDHE_ECDSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Ja	Ja	Nein
ECDHE_ECDSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Ja	Nein	Ja
ECDHE_RSA_AES_128_GCM_SHA256 [b]	TLS 1.2	AEAD AES-128 GCM	AES	128	Ja	Nein	Nein
ECDHE_RSA_AES_256_GCM_SHA384 [b]	TLS 1.2	AEAD AES-256 GCM	AES	256	Ja	Nein	Nein
TLS_RSA_WITH_NULL_SHA256 [b]	TLS 1.2	SHA-256	--	0	Nein	Nein	Nein
ECDHE_RSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	--	0	Nein	Nein	Nein
ECDHE_ECDSA_NULL_SHA256 [b]	TLS 1.2	SHA-1	--	0	Nein	Nein	Nein
TLS_RSA_WITH_NULL_NULL [b]	TLS 1.2	--	--	0	Nein	Nein	Nein
TLS_RSA_WITH_RC4_128_SHA256 [b]	TLS 1.2	SHA-1	RC4	128	Nein	Nein	Nein

CipherSpec-Name	Verwendetes Protokoll	MAC-Algorithmus	Verschlüsselungsalgorithmus	Verschlüsselungsbits	FIPS [1]	Suite B mit 128 Bit	Suite B mit 192 Bit

Anmerkungen:

1. 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.
2. 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.
3. Die Größe des Handshakeschlüssels beträgt 1024 Bit.
4. Mithilfe dieser Verschlüsselungsspezifikation (CipherSpec) kann eine Verbindung von WebSphere MQ Explorer zu einem Warteschlangenmanager nicht geschützt werden, es sei denn, für die vom Explorer verwendete JRE gelten die entsprechenden uneingeschränkten Richtliniendateien.
5. Diese CipherSpec wurde vor dem 19. Mai 2007 FIPS 140-2-zertifiziert.
6. 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.
7. Mit dieser CipherSpec können bis zu 32 GB Daten übertragen werden, bevor die Verbindung mit Fehler AMQ9288 beendet wird. Um diesen Fehler zu vermeiden, sollten Sie Triple DES nicht verwenden oder bei Verwendung dieser CipherSpec die Rücksetzung der geheimen Schlüssel ermöglichen.

Plattformunterstützung:

- a Auf allen unterstützten Plattformen verfügbar.
- b Nur auf UNIX, Linux, and Windows-Plattformen verfügbar.

Zugehörige Konzepte

„Digitale Zertifikate und CipherSpec -Kompatibilität in IBM WebSphere MQ” auf Seite 36
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 WebSphere MQ erläutert.

Zugehörige Verweise

CHANNEL DEFINE CHANNEL

ALTER CHANNEL

Informationen zu CipherSpecs mit IBM WebSphere MQ Explorer abrufen

Sie können IBM WebSphere MQ Explorer verwenden, um Beschreibungen von CipherSpecs anzuzeigen.

Gehen Sie wie folgt vor, um Informationen zu den CipherSpecs in „CipherSpecs angeben” auf Seite 230 abzurufen:

1. Öffnen Sie den **IBM WebSphere 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.

Alternativen für die Angabe von CipherSpecs

Für Plattformen, auf denen SSL-Unterstützung vom Betriebssystem bereitgestellt wird, werden eventuell auch neue CipherSpecs unterstützt. Sie können eine neue CipherSpec mit dem Parameter SSLCIPH angeben, aber der von Ihnen angegebene Wert hängt von Ihrer Plattform ab.

Anmerkung: Dieser Abschnitt gilt nicht für UNIX-, Linux -oder Windows -Systeme, da die CipherSpecs mit dem Produkt WebSphere MQ bereitgestellt werden, sodass neue CipherSpecs nach dem Versand nicht verfügbar sind.

Auf Plattformen, auf denen die SSL-Unterstützung vom Betriebssystem zur Verfügung gestellt wird, werden eventuell auch neue CipherSpecs unterstützt, die nicht in „CipherSpecs angeben" auf Seite 230 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 Angabe einer SSL-CipherSpec entsprechen, die für die SSL-Version auf Ihrem System gültig ist und von ihr unterstützt wird.

IBM i

Eine Zeichenfolge mit zwei Zeichen, die einen Hexadezimalwert darstellt.

Weitere Informationen zu den zulässigen Werten finden Sie in der entsprechenden Produktdokumentation (suchen Sie nach *cipher_spec* in der IBM i -Produktdokumentation).

Sie können entweder den Befehl CHGMQMCHL oder den Befehl CRTMQMCHL verwenden, um den Wert anzugeben, z. B.:

```
CRTMQMCHL CHLNAME('channel name') SSLCIPH('hexadecimal value')
```

Sie können auch den MQSC-Befehl ALTER QMGR verwenden, um den Parameter SSLCIPH festzulegen.

z/OS

Eine Zeichenfolge mit zwei Zeichen, die einen Hexadezimalwert darstellt. Die hexadezimalen Codes entsprechen den im SSL-Protokoll definierten Werten.

Weitere Informationen finden Sie in der Beschreibung von gsk_environment_open () im API-Referenzkapitel von *z/OS Cryptographic Services System SSL Programming*, SC24-5901. Dort finden Sie eine Liste aller unterstützten SSL V3.0 -und TLS V1.0 -Verschlüsselungsspezifikationen in Form von zweistelligen hexadezimalen Codes.

Hinweise zu WebSphere MQ -Clustern

Bei WebSphere MQ -Clustern ist es am sichersten, die CipherSpec -Namen in „CipherSpecs angeben" auf Seite 230zu verwenden. 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 und Cluster" auf Seite 263.

CipherSpec für einen IBM WebSphere MQ MQI-Client angeben

Sie haben drei Optionen zur Angabe einer CipherSpec für einen IBM WebSphere 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.
- Active Directory verwenden (auf Windows -Systemen mit Active Directory -Unterstützung)

CipherSuite mit IBM WebSphere MQ -Klassen für Java und IBM WebSphere MQ -Klassen für JMS angeben

IBM WebSphere MQ -Klassen für Java und IBM WebSphere MQ -Klassen für JMS geben CipherSuites anders an als andere Plattformen.

Informationen zur Angabe einer CipherSuite mit IBM WebSphere MQ -Klassen für Java finden Sie im Abschnitt Unterstützung für Secure Sockets Layer (SSL).

Informationen zur Angabe einer CipherSuite mit IBM WebSphere MQ Classes for JMS finden Sie im Abschnitt Secure Sockets Layer (SSL) mit WebSphere MQ Classes for JMS.

Prüfprotokollierungs

Sie können mithilfe von Ereignisnachrichten auf Sicherheitseinbrüche oder unbefugte Zugriffe überprüfen. Sie können die Sicherheit Ihres Systems auch mit dem IBM WebSphere 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 im Abschnitt Warteschlangenmanagerereignisse. 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 SSL 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. Berechtigen Sie

einen fernen Warteschlangenmanager, indem Sie die Benutzer-ID in der Nachricht unter z/OSmit RACF oder auf anderen Plattformen mit OAM überprüfen.

Informationen zu diesem Vorgang

Verwenden Sie die Sicherheitsfunktionen einer Plattform und den Zugriffssteuerungsmechanismus in WebSphere 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 in WebSphere MQ für 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/OSkann die Überprüfung sowohl die aus dem Netz empfangene als auch die aus MCAUSERabgeleitete Benutzer-ID umfassen.

   CTX
   Verwenden Sie die Benutzer-ID in den Kontextinformationen, die der Nachricht zugeordnet sind. Unter z/OS kann die Prüfung die Verwendung der aus dem Netz empfangenen und/oder von MCAUSERabgeleiteten Benutzer-ID beinhalten. Verwenden Sie diese Option, wenn der Link vertrauenswürdig und authentifiziert ist.

   ONLYMCA (nurz/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 (nurz/OS)
   Wie bei CTX , aber jede aus dem Netz empfangene Benutzer-ID wird nicht verwendet.

Berechtigung zum Einreihen von Nachrichten in ferne Clusterwarteschlangen berechtigen

Autorisieren Sie auf Ihrer Plattform den Zugriff, um eine Verbindung zum Warteschlangenmanager herzustellen und in die Warteschlange auf diesem Warteschlangenmanager einzureihen.

Informationen zu diesem Vorgang

Das Standardverhalten ist die Ausführung der Zugriffssteuerung für SYSTEM.CLUSTER.TRANSMIT.QUEUE. Beachten Sie, dass dieses Verhalten auch dann gilt, wenn Sie mehrere Übertragungswarteschlangen verwenden.

Das in diesem Abschnitt beschriebene spezifische Verhalten gilt nur, wenn Sie das Attribut **ClusterQueueAccessControl** in der Datei qm.ini als *RQMName* konfiguriert haben, wie in der Sicherheitszeilengruppe beschrieben, und den Warteschlangenmanager erneut gestartet haben.

Prozedur

- Geben Sie unter UNIX, Linux und Windows die folgenden Befehle aus:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
```

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.

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 204 befolgen.

Zugehörige Informationen
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 auf Basis folgender Kriterien blockieren: der fernen IP-Adresse, dem Namen des fernen Warteschlangenmanagers oder dem vom fernen System bereitgestellten SSL/TLS-DN.
- 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.

   Windows UNIX Linux **Windows, UNIX and Linux -Systeme**

   - 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.

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)
```

Ergebnisse

Der Warteschlangenmanager, dessen Entfernung erzwungen wird, ändert sich nicht: seine lokalen Clusterdefinitionen zeigen, dass er sich im Cluster befindet. 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 und Cluster

Achten Sie beim Konfigurieren von SSL für Cluster darauf, dass eine CLUSRCVR-Kanaldefinition an andere Warteschlangenmanager als automatisch definierter CLUSSDR-Kanal weitergegeben wird. Wenn ein CLUSRCVR-Kanal SSL verwendet, müssen Sie SSL auf allen Warteschlangenmanagern konfigurieren, die mit dem Kanal kommunizieren.

Weitere Informationen zu SSL finden Sie unter WebSphere MQ -Unterstützung für SSL und TLS. Die Empfehlung gibt es im Allgemeinen für Clusterkanäle, aber Sie können die folgenden Hinweise beachten:

In einem IBM WebSphere 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 SSL-Konnektivität konfiguriert ist, gelten folgende Hinweise:

- Alle Warteschlangenmanager, die mit diesem CLUSRCVR kommunizieren möchten, müssen Zugriff auf SSL-Unterstützung haben. Diese SSL-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 konfigu-

riert 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 WebSphere MQ Development' definiert ist. Alle haben allgemeine Namen im Format DEVT.QMnnn, wobei nnn für einen numerischen Wert steht.

In diesem Fall lässt der SSLPEER -Wert C=UK, O=IBM, OU=WebSphere MQ Development, CN=DEVT.QM* auf dem CLUSRCVR zu, dass alle erforderlichen Clustersenderkanäle erfolgreich eine Verbindung herstellen, 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 Zeichenfolge CipherSpec RC4_SHA_US einen Wert von 05 in IBM i hat, aber keine gültige Spezifikation für UNIX-, Linux -oder Windows -Systeme ist. Wenn angepasste SSLCIPH-Parameter in einem CLUSRCVR verwendet werden, sollten sich daher alle resultierenden automatisch definierten Clustersenderkanäle auf Plattformen befinden, auf denen die zugrunde liegende SSL-Unterstützung diese CipherSpec implementiert und auf denen 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 nach Möglichkeit die Textzeichenfolgen der CipherSpec (z. B. RC4_MD5_US).

Ein Parameter SSLCRLNL gilt für einen einzelnen WS-Manager und wird nicht an andere Warteschlangenmanager innerhalb eines Clusters weitergegeben.

Upgrade für Clusterwarteschlangenmanager und -kanäle auf SSL 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 möglicherweise neu im aktuellen WebSphere 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.

Führen Sie für alle Warteschlangenmanager im Cluster ein Upgrade auf WebSphere MQ V6 oder höher durch, wenn sie diese Versionen noch nicht aufweisen. Verteilen Sie die Zertifikate und Schlüssel so, dass SSL von jedem von ihnen funktioniert.

Informationen zu diesem Vorgang

Ändern Sie jeweils einen CLUSRCVR und lassen Sie die Änderungen durch den Cluster fließen, bevor Sie den nächsten ändern. Stellen Sie sicher, dass Sie den Reverse-Pfad erst ändern, wenn die Änderungen für den aktuellen Kanal im gesamten Cluster verteilt wurden.

Vorgehensweise

1. Schalten Sie die CLUSRCVR -Kanäle in beliebiger Reihenfolge auf SSL um.

   Die Änderungen fließen in die entgegengesetzte Richtung über Kanäle, die nicht in SSL geändert werden.
2. Stellen Sie alle manuellen CLUSSDR-Kanäle auf SSL 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 der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle interessierten Warteschlangenmanager hochladen, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.

Zugehörige Konzepte

„CipherSpecs angeben” auf Seite 230
Geben Sie eine CipherSpec an, indem Sie den Parameter **SSLCIPH** im MQSC-Befehl **DEFINE CHANNEL** oder im MQSC-Befehl **ALTER CHANNEL** verwenden.

„Digitale Zertifikate und CipherSpec -Kompatibilität in IBM WebSphere MQ” auf Seite 36
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 WebSphere MQ erläutert.

Zugehörige Informationen

Clustering: Best Practices für REFRESH CLUSTER verwenden

SSL oder TLS auf Clusterwarteschlangenmanagern und Kanälen inaktivieren

Wenn Sie SSL oder TLS inaktivieren möchten, 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 .

   Sie können SSL oder TLS auf den Clusterempfängerkanälen in beliebiger Reihenfolge inaktivieren.

   Beachten Sie, dass die Änderungen in die entgegengesetzte Richtung über Kanäle fließen, auf denen Sie SSL oder TLS aktiv lassen.
2. Überprüfen Sie, ob der neue Wert in allen anderen Queue Managern über den Befehl **DISPLAY CLUSQMGR(*)** ALL widergespiegelt wird.
3. Inaktivieren Sie SSL oder TLS auf allen manuellen Clustersenderkanälen.

   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 finden Sie unter Aktualisierung in einem großen Cluster 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 26 auf Seite 266 dargestellt und im folgenden Beispiel beschrieben.

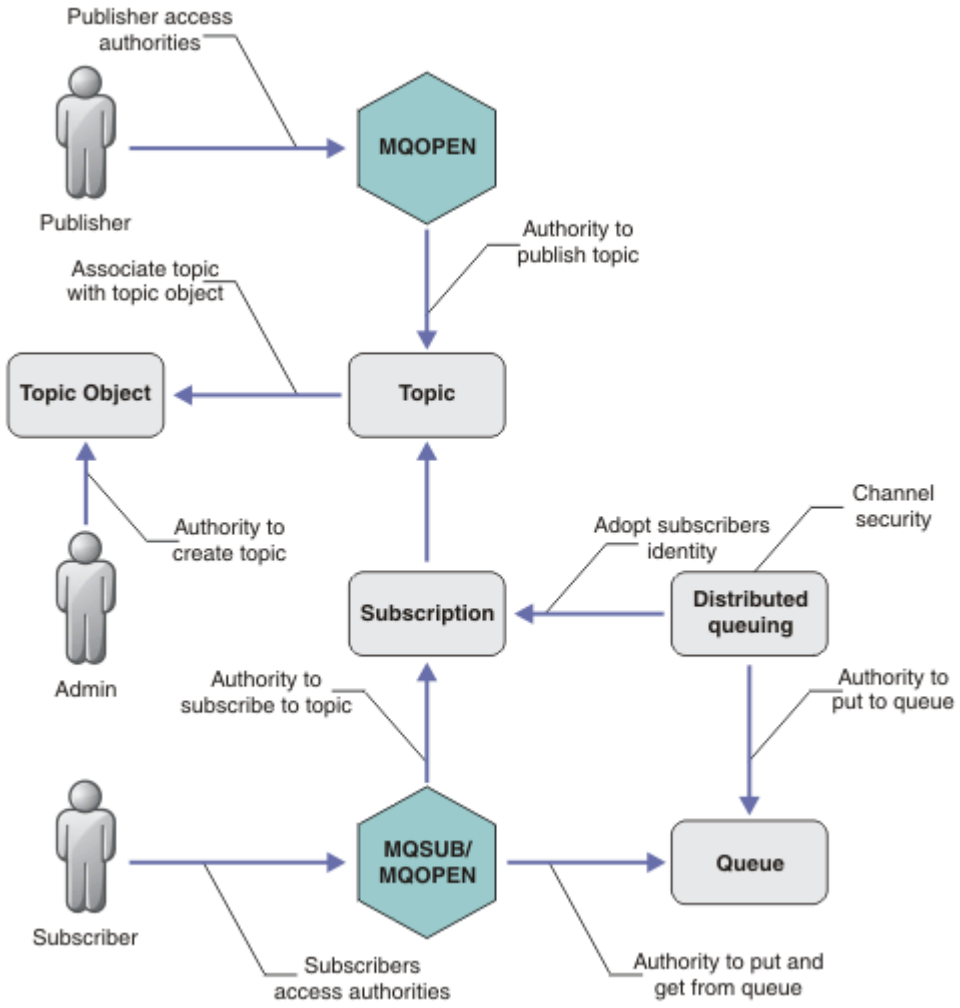


Abbildung 26. Publish/Subscribe-Sicherheitsbeziehungen

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.„Thema Sicherheitsmodell” auf Seite 268 erläutert, wie Themen mithilfe von Themenobjekten gesichert werden.

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

273 und „Einem Benutzer den Zugriff auf die Veröffentlichung in einem Thema gewähren” auf Seite 279.

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 270

Subskription mit einem Thema
„Abonnieren mit einer Themenzeichenfolge, in der der Themenknoten nicht vorhanden ist” auf Seite 270

Subskription unter Verwendung eines Themas mit Platzhalterzeichen
„Subskription mit einer Themenzeichenfolge, die Platzhalterzeichen enthält” auf Seite 271

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. Weitere Informationen finden Sie im Abschnitt „Subskriptionssicherheit” auf Seite 283.

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. Weitere Informationen finden Sie im Abschnitt „Berechtigung für Zielwarteschlangen” auf Seite 271.

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

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 WebSphere MQ -Clusterkonfigurationen sind Kanäle mit PUTAUT=DEF konfiguriert, um Nachrichten mit der Berechtigung des Kanalprozesses in Zielwarteschlangen zu stellen. Sie können die Kanalkonfiguration so ändern, dass PUTAUT=CTX verwendet wird, damit der subskribierende Benutzer über die Berechtigung zur Weitergabe einer Subskription an einen anderen WS-Manager in einem Cluster verfügt.

Unter Publish/Subscribe-Sicherheit zwischen Warteschlangenmanagern wird beschrieben, wie Sie Ihre Kanaldefinitionen ändern, um zu steuern, wer 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. WebSphere 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 Ver-

öffentlichungsoperation durchgeführt wird, basiert die erteilte Berechtigung auf den Sicherheitsattributen des zugeordneten Themenbaumknotens.

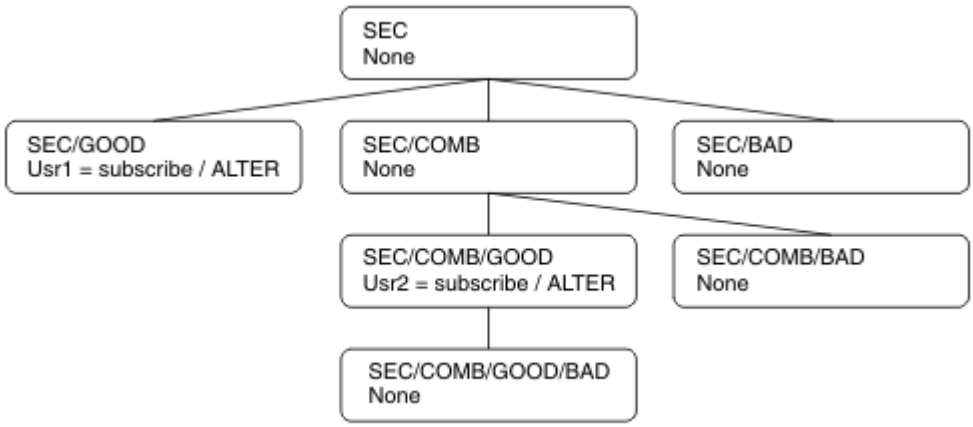
Die Sicherheitsattribute sind eine Zugriffssteuerungsliste, die angibt, welche Berechtigung eine bestimmte Betriebssystembenutzer-ID oder Sicherheitsgruppe für das Themenobjekt hat.

Betrachten Sie das folgende Beispiel, in dem die Themenobjekte mit den Sicherheitsattributen oder den angezeigten Berechtigungen definiert wurden:

Tabelle 17. Beispiel-Topic-Objektberechtigungen

Themenname	**Themenzeichenfolge**	**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

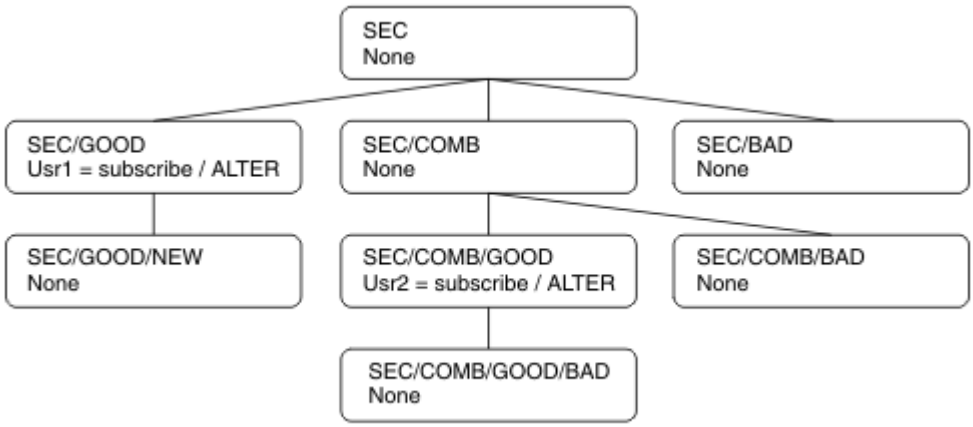
Das Verhalten ist dasselbe wie bei der Angabe des Themas durch den Namen des MQCHAR48-Objekts.

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 Baum-

struktur 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

Wenn Sie ein Thema subskribieren, ist einer der Parameter der Handle hobj einer Warteschlange, die für die Ausgabe geöffnet wurde, um die Veröffentlichungen zu empfangen.

Wenn hobj nicht angegeben wird, aber leer ist, wird eine verwaltete Warteschlange erstellt, wenn die folgenden Bedingungen zutreffen:

- Die Option MQSO_MANAGED wurde angegeben.
- Die Subskription ist nicht vorhanden.
- Erstellen ist angegeben.

Wenn hobj leer ist und Sie eine vorhandene Subskription 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 die Prüfungen *nicht* in der zugrunde liegenden Warteschlange überprüft. 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 18 auf Seite 273):

Tabelle 18. Benutzer-IDs, die für Sicherheitsprüfungen für Befehle verwendet werden

Befehl	SUBUSER angegeben und leer	SUBUSER angegeben und abgeschlossen	SUBUSER nicht angegeben
	Administrator-ID verwenden		Administrator-ID verwenden
	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, in dem eine Zugriffssteuerung auf Themen so konfiguriert ist, dass die Sicherheitssteuerung je nach Bedarf angewandt werden kann.

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.

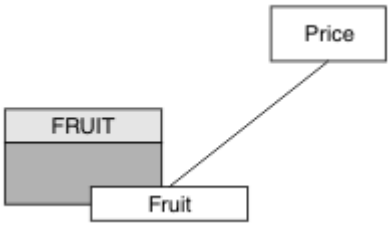


Abbildung 27. Zugriffsbeispiel für Themenobjektzugriff

Tabelle 19. Beispielthemenobjektzugriff		
Thema	**Subskriptions-zugriff erforder-lich**	**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:
   - Andere Plattformen:

     Erteilen Sie den Zugriff auf USER1 , um das Thema `"Price/Fruit"` zu subskribieren, indem Sie dem Benutzer Zugriff auf das Objekt FRUIT erteilen. Führen Sie dazu den Berechtigungsbefehl für die Plattform aus:

     Windows UNIX Linux **Windows, UNIX and Linux -Systeme**

     ```
     setmqaut -t topic -n FRUIT -p USER1 +sub
     ```

Ergebnisse

Wenn USER1 versucht, das Thema `"Price/Fruit"` zu subskribieren, ist das Ergebnis erfolgreich.

Wenn USER2 versucht, das Thema `"Price/Fruit"` zu subskribieren, schlägt das Ergebnis mit einer `MQRC_NOT_AUTHORIZED` -Nachricht zusammen mit Folgendem fehl:

- Windows UNIX Linux 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"
  ```

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 273 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.

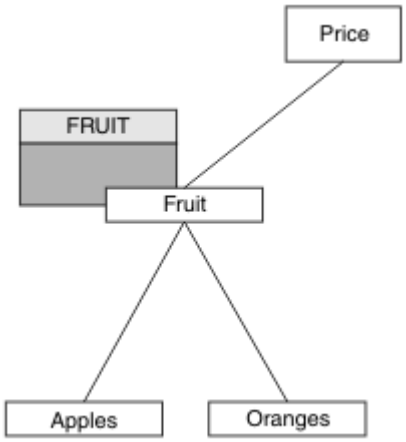


Abbildung 28. Beispiel für das Erteilen des Zugriffs auf ein Thema in einer Themenstruktur

Tabelle 20. Zugriffsvoraussetzungen für Beispielthemen und Themenobjekte

Thema	Subskriptionszugriff erforderlich	Themenobjekt
Preis	Kein Benutzer	--
Preis/>Obst	USER1	OBST
Preis/Obst/Äpfel	USER1	
Preis/Obst/Orangen	USER1	

In der vorherigen Task wurde USER1 der Zugriff zum Subskribieren des Themas `"Price/Fruit"` erteilt, indem ihm der Zugriff auf das Profil `hlq.SUBSCRIBE.FRUIT` unter z/OS und der Zugriff auf das Profil FRUIT 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/OSwerden in der Konsole die folgenden Nachrichten angezeigt, die den vollständigen Sicherheitspfad durch die Themenstruktur anzeigen, die versucht wurde:

```
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 unter z/OS empfangen, sind identisch mit den Nachrichten, die Sie in der vorherigen Task empfangen haben, 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 274 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.

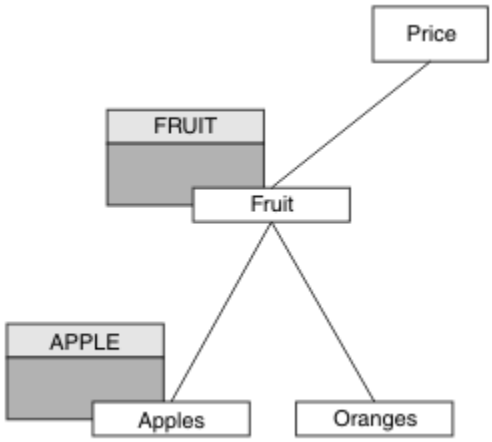


Abbildung 29. Zugriff auf bestimmte Themen in einer Themenstruktur erteilen

Tabelle 21. Zugriffsvoraussetzungen für Beispielthemen und Themenobjekte

Thema	Subskriptionszugriff erforderlich	Themenobjekt
Preis	Kein Benutzer	--
Preis/>Obst	USER1	OBST
Preis/Obst/Äpfel	BENUTZER1 und BENUTZER2	APFEL

Tabelle 21. Zugriffsvoraussetzungen für Beispielthemen und Themenobjekte (Forts.)		
Thema	**Subskriptions-zugriff erforderlich**	**Themenobjekt**
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:
   - 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:

     Windows UNIX Linux **Windows, UNIX and Linux -Systeme**

     ```
     setmqaut -t topic -n APPLE -p USER2 +sub
     ```

Ergebnisse

Wenn unter z/OS USER1 versucht, 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 jedoch eine Nachricht RACF `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:

- Windows UNIX Linux Auf Windows-, UNIX-und Linux -Plattformen 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 unter z/OSzusätzliche `ICH` -Nachrichten in der Konsole empfangen. Sie können dies vermeiden, wenn Sie die Themenstruktur auf eine andere Weise sichern.

Zugriffssteuerung ändern, um zusätzliche Nachrichten zu vermeiden

Dieser Abschnitt ist der vierte in einer Liste von Tasks, die Ihnen mitteilen, wie Sie Zugriff auf die Subskription von Topics durch mehrere Benutzer erteilen und zusätzliche RACF ICH408I -Nachrichten unter z/OSvermeiden können.

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 276 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.

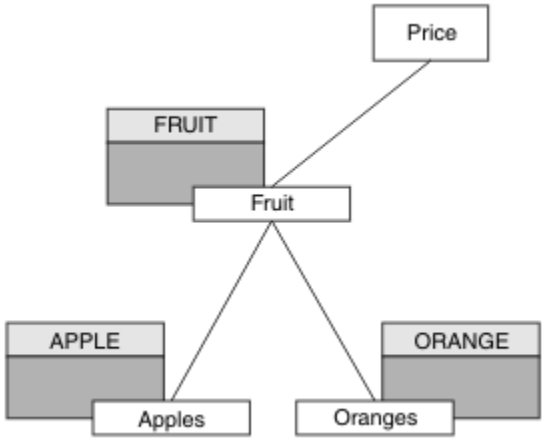


Abbildung 30. Beispiel für das Erteilen der Zugriffssteuerung, um zusätzliche Nachrichten zu vermeiden.

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:
   - Andere Plattformen:

     Richten Sie den entsprechenden Zugriff mithilfe der Berechtigungsbefehle für die Plattform ein:

     Windows, UNIX and Linux -Systeme

     ```
     setmqaut -t topic -n ORANGE -p USER1 +sub
     setmqaut -t topic -n APPLE -p USER1 +sub
     ```

Ergebnisse

Wenn USER1 unter z/OSversucht, das Thema `"Price/Fruit/Apples"` zu subskribieren, ist die erste Sicherheitsprüfung im `hlq.SUBSCRIBE.APPLE` -Profil 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:

- Windows UNIX Linux 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"
```

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:

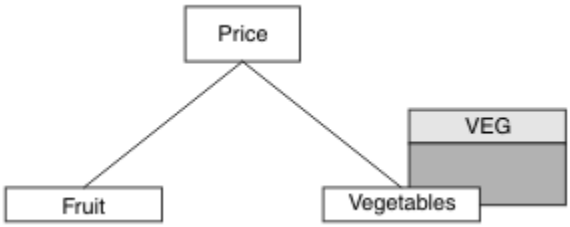


Abbildung 31. Publizierungszugriff auf ein Thema erteilen

Tabelle 22. 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:
   - 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:

     Windows UNIX Linux **Windows, UNIX and Linux -Systeme**

     ```
     setmqaut -t topic -n VEG -p USER1 +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:

- Windows UNIX Linux 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"
```

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 279 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.

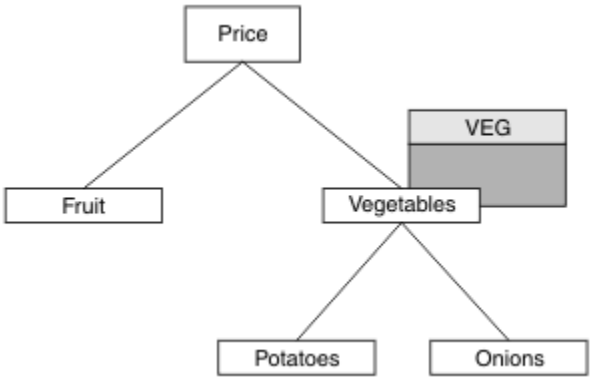


Abbildung 32. Publizierungszugriff auf ein Thema in einer Themenstruktur erteilen

Tabelle 23. Beispiel für Veröffentlichungszugriffs-Anforderungen

Thema	Subskriptionszugriff erforderlich	Themenobjekt
Preis	Kein Benutzer	--
Preis/Gemüse	USER1	VEG

Tabelle 23. Beispiel für Veröffentlichungszugriffs-Anforderungen (Forts.)		
Thema	**Subskriptions-zugriff erforderlich**	**Themenobjekt**
Preis/Gemüse/Kartoffeln	USER1	
Preis/Gemüse/Zwiebeln	USER1	

In der vorherigen Task wurde USER1 Zugriff auf das Veröffentlichungsthema `"Price/Vegetables/Potatoes"` erteilt, indem der Zugriff auf das `hlq.PUBLISH.VEG` -Profil unter z/OS oder auf das `VEG` -Profil 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/OSwerden in der Konsole die folgenden Nachrichten angezeigt, die den vollständigen Sicherheitspfad durch die Themenstruktur anzeigen, die versucht wurde:

```
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 unter z/OS empfangen, sind identisch mit den Nachrichten, die Sie in der vorherigen Task empfangen haben, 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 280 beschriebene Konfiguration verwendet.

Informationen zu diesem Vorgang

In einer vorherigen Aufgabe wurde USER1 Zugriff zum Subskribieren des Themas `"Price/Fruit"`erteilt. In diesem Abschnitt erfahren Sie, wie Sie diesem Benutzer den Zugriff auf die Veröffentlichung zu diesem Thema gewähren.

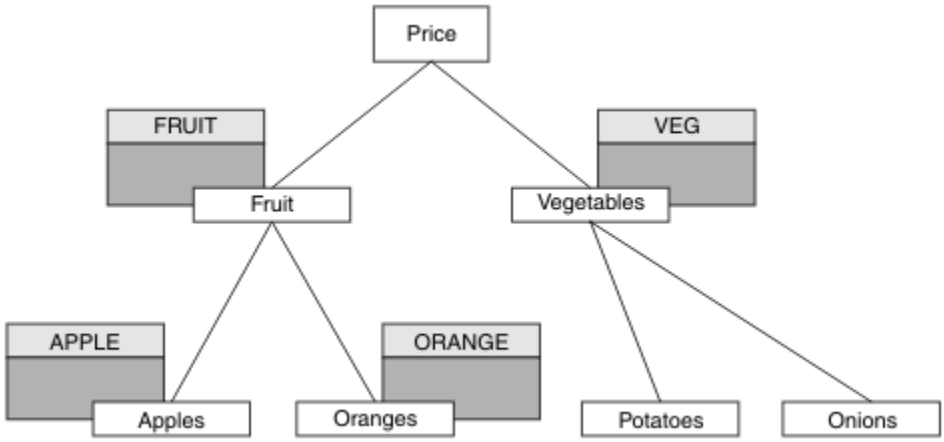


Abbildung 33. Zugriff für Veröffentlichung und Subskribierung erteilen

Tabelle 24. Beispiel für Veröffentlichungs-und Subskribierungszugriffanforderungen

Thema	Subskription-szugriff 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:

- 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:

  Windows UNIX Linux **Windows, UNIX and Linux -Systeme**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +pub
  ```

Ergebnisse

Unter z/OSwird die Sicherheitsprüfung beim MQOPEN-Aufruf bestanden, wenn USER1 versucht, im Thema "Price/Fruit" zu veröffentlichen.

Wenn USER2 versucht, beim Thema "Price/Fruit" zu veröffentlichen, schlägt das Ergebnis mit einer MQRC_NOT_AUTHORIZED -Nachricht zusammen mit Folgendem fehl:

- Windows UNIX Linux Auf Windows-, UNIX-und Linux -Plattformen 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 25. 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 verschiedenen Ebenen innerhalb der Themenstruktur haben, stellt eine sorgfältige Planung sicher, dass Sie keine zusätzlichen 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 26. 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.

IBM WebSphere MQ Advanced Message Security

IBM WebSphere MQ Advanced Message Security (AMS) ist eine separat lizenzierte Komponente von IBM WebSphere MQ Advanced Message Security , die ein hohes Maß an Schutz für sensible Daten bietet, die durch das IBM WebSphere MQ Advanced Message Security -Netz fließen, ohne die Endanwendungen zu beeinträchtigen.

Die Übersicht IBM WebSphere MQ Advanced Message Security

IBM WebSphere MQ-Anwendungen können IBM WebSphere MQ 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-Rückkehrcodes in IBM WebSphere MQ AMS -Nachrichten

Geändertes Verhalten zwischen Version 7.0.1 und Version 7.5

Da IBM Erweiterte Nachrichtensicherheit in WebSphere MQ 7.5eine Komponente wurde, haben sich einige Aspekte der IBM WebSphere MQ AMS -Funktionalität geändert, die sich auf vorhandene Anwendungen, Verwaltungsscripts oder Managementprozeduren auswirken können.

Lesen Sie die folgende Liste der Änderungen sorgfältig durch, bevor Sie ein Upgrade der Warteschlangenmanager auf Version 7.5durchführen. Entscheiden Sie, ob Sie Änderungen an vorhandenen Anwendungen, Scripts und Prozeduren planen müssen, bevor Sie mit der Migration von Systemen auf IBM WebSphere MQ Version 7.5:

- Die IBM WebSphere MQ AMS -Installation ist Teil des WebSphere MQ -Installationsprozesses.
- IBM WebSphere MQ AMS -Sicherheitsfunktionen werden mit der Installation aktiviert und mit Sicherheitsrichtlinien gesteuert. Sie müssen keine Interceptor aktivieren, damit IBM WebSphere MQ AMS -Startdaten abfangen können.
- IBM WebSphere MQ AMS in WebSphere MQ Version 7.5 erfordert nicht die Verwendung des Befehls **cfgmqs** wie in der eigenständigen Version von IBM WebSphere MQ AMS.

Features und Funktionen von IBM WebSphere MQ Advanced Message Security

Erweiterte Nachrichtensicherheit erweitert die Sicherheitsservices von WebSphere MQ , um Datensignatur und -verschlüsselung 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 IBM WebSphere MQ AMS sicher, dass ein Sender von Nachrichtendaten berechtigt ist, signierte Nachrichten in eine Zielwarteschlange zu stellen.

Es folgt eine vollständige Liste der IBM WebSphere MQ AMS -Funktionen:

- Sichert sensible oder hochwertige Transaktionen, die von WebSphere 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.
- Schützt vorhandene proprietäre und vom Kunden geschriebene Anwendungen für WebSphere MQ.

Fehlerbehandlung

Erweiterte Nachrichtensicherheit 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-Headerfehler-Auf den WebSphere MQ AMS-Nachrichtenheader 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.

WebSphere MQ AMS verwendet das SYSTEM.PROTECTION.ERROR.QUEUE als Fehlerbehandlungswarteschlange. Alle Nachrichten, die von IBM WebSphere MQ AMS in SYSTEM.PROTECTION.ERROR.QUEUE steht ein MQDLH-Header.

Ihr WebSphere MQ -Administrator kann auch das SYSTEM.PROTECTION.ERROR.QUEUE als Aliaswarteschlange, die auf eine andere Warteschlange verweist.

Zentrale Konzepte

Informieren Sie sich über die Schlüsselkonzepte in Erweiterte Nachrichtensicherheit , um zu verstehen, wie das Tool funktioniert und wie es effektiv verwaltet werden kann.

PKI-Infrastruktur

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. Erweiterte Nachrichtensicherheit 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 verwenden” auf Seite 311
Um für WebSphere MQ -Anwendungen einen transparenten Verschlüsselungsschutz bereitzustellen, verwendet Erweiterte Nachrichtensicherheit die Schlüsselspeicherdatei, in der öffentliche Schlüsselzertifikate und ein privater Schlüssel gespeichert werden.

Digitale Zertifikate

Erweiterte Nachrichtensicherheit 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: Erweiterte Nachrichtensicherheit unterstützt selbst signierte Zertifikate in Java-und nativen Anwendungen.

Zugehörige Konzepte

„Kryptografie” auf Seite 7
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* .

Objektberechtigungsmanager

Der Object Authority Manager (OAM) ist die Berechtigungsservicekomponente, die mit den WebSphere MQ -Produkten bereitgestellt wird.

Der Zugriff auf Erweiterte Nachrichtensicherheit-Entitäten wird über WebSphere 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 Erweiterte Nachrichtensicherheit-Objekte über 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

Unterstützte Technologie

Erweiterte Nachrichtensicherheit hängt von mehreren IT-Komponenten ab, mit denen eine Sicherheitsinfrastruktur bereitgestellt wird.

Erweiterte Nachrichtensicherheit unterstützt die folgenden WebSphere MQ APIs (Application Programming Interfaces, Anwendungsprogrammierschnittstellen):

- Nachrichtenwarteschlangenschnittstelle (MQI)
- WebSphere MQ Java Message Service (JMS) 1.0.2 und 1.1.
- WebSphere MQ -Basisklassen für Java
- WebSphere MQ-Klasse für .Net in einem nicht verwalteten Modus

Anmerkung: Erweiterte Nachrichtensicherheit unterstützt X.509-konforme Zertifizierungsstellen.

Bekannte Einschränkungen

Informationen zu Einschränkungen von IBM WebSphere MQ Advanced Message Security.

- Die folgenden IBM WebSphere MQ -Optionen werden nicht unterstützt:
  - Publish/Subscribe.
  - Kanaldatenkonvertierung.
  - Verteilerlisten.
  - Anwendungsnachrichtensegmentierung
  - Die Verwendung von Anwendungen, die keine Thread-Anwendungen sind, mit dem API-Exit auf HP-UX-Plattformen.
  - IBM WebSphere MQ -Klassen für .NET in einem verwalteten Modus (Client-oder Bindungsverbindungen).
  - Message Service Client für .NET-Anwendungen (XMS).
  - Nachrichtenservice-Client für C/C++-Anwendungen (XMS supportPac IA94).
- Alle Java -Anwendungen sind von IBM Java Runtime abhängig.

  IBM WebSphere MQ Advanced Message Security unterstützt keine JRE, die von anderen Anbietern bereitgestellt wird.

- JMS-und Java -Clientanwendungen verwenden IBM WebSphere MQ Advanced Message Security im Clientmodus.

  Jede JMS-oder Java-Clientanwendung (einschließlich IBM WebSphere MQ Explorer -und IBM WebSphere MQ Managed File Transfer -Agenten) kann IBM WebSphere MQ Advanced Message Security nicht im Clientmodus mit einem WebSphere MQ -Warteschlangenmanager vor Version 7.5verwenden.

  Um Nachrichtenschutzrichtlinien zu verwenden, müssen diese Anwendungen entweder mit einem IBM WebSphere MQ Version 7.5 -Warteschlangenmanager interagieren oder im lokalen Bindungsmodus eine Verbindung zu einem Warteschlangenmanager auf demselben System wie die Anwendung herstellen.
- Sie sollten es vermeiden, zwei oder mehr Zertifikate mit denselben definierten Namen in einer einzigen Keystore-Datei zu speichern, da die Funktionsweise des IBM WebSphere MQ Advanced Message Security -Intereceptors mit solchen Zertifikaten nicht definiert ist.
- Der IBM WebSphere MQ Version 7.5 -Ressourcenadapter unterstützt IBM WebSphere MQ Advanced Message Securitynicht. Wenn Nachrichtenschutz mit IBM WebSphere MQ classes for JMS -oder IBM WebSphere MQ classes for Java -Anwendungen verwendet werden muss, die in einer Anwendungsserverumgebung ausgeführt werden, gilt Folgendes:
  - Der Anwendungsserver muss für die Verwendung des Ressourcenadapters Version 8.0 oder höher konfiguriert sein.
  - Andernfalls muss das Abfangen des Nachrichtenkanalagenten (MCA) verwendet werden.

Benutzerszenarios

Machen Sie sich mit möglichen Szenarios vertraut, um die Geschäftsziele zu verstehen, die Sie mit Erweiterte Nachrichtensicherheit erreichen können.

Schnelleinstieg für Windows -Plattformen

Verwenden Sie dieses Handbuch, um IBM Erweiterte Nachrichtensicherheit schnell für die Nachrichtensicherheit auf Windows -Plattformen zu konfigurieren. 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 WebSphere MQ-Funktionen für Windows-Systeme.

Informationen zur Verwendung des Befehls **setmqenv** zum Initialisieren der aktuellen Umgebung, damit die entsprechenden WebSphere MQ -Befehle vom Betriebssystem lokalisiert und ausgeführt werden können, finden Sie im Abschnitt setmqenv.

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. Erweiterte Nachrichtensicherheit verwendet Interceptors zum Signieren und Verschlüsseln von Nachrichten an dem Punkt, an dem sie über die Standardschnittstelle von WebSphere MQ in die WebSphere MQ -Infrastruktur gelangen. Die Basiseinrichtung wird in WebSphere MQ vorgenommen und in den folgenden Schritten konfiguriert.

Sie können WebSphere 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 \WebSphere MQ\binverwenden. 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 zu dem 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 müssen Sie es den beiden Benutzern ermöglichen, die Systemrichtlinienwarteschlange zu durchsuchen und Nachrichten in die Fehlerwarteschlange zu stellen.

   ```
   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
   ```

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 293 beschrieben.

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 mithilfe von 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 „Schnelleinstieg für Java-Clients" auf Seite 300). Für alle anderen Sprachen und für Java-Anwendungen, die lokale Bindungen verwenden, sind die Schritte in diesem Handbuch korrekt.

Vorgehensweise

1. Verwenden Sie die grafische Benutzerschnittstelle von IBM Key Management (`strmqikm.exe`), um eine neue Schlüsseldatenbank für den Benutzer `alice`zu erstellen.

   ```
   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. In diesem Szenario werden selbst signierte Zertifikate 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

Sie müssen Erweiterte Nachrichtensicherheit -Interceptors auf das Verzeichnis verweisen, in dem sich die Schlüsseldatenbanken und Zertifikate located.This geschieht über die Datei `keystore.conf` , die diese Informationen in Klartext enthält. Jeder Benutzer muss über eine separate `keystore.conf` -Datei 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 von `keystore.conf` wie folgt angezeigt:

```
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.
- Die Zertifikatsbezeichnung kann Leerzeichen enthalten, also "Alice_Cert" und "Alice_Cert". werden beispielsweise als Bezeichnungen von zwei verschiedenen Zertifikaten erkannt. Um Unklarheiten zu vermeiden, ist es jedoch besser, keine Leerzeichen im Namen der Bezeichnung zu verwenden.
- Es gibt folgende Keystore-Formate: `CMS` (Syntax für verschlüsselte Nachrichten), `JKS` (Java-Keystore) und `JCEKS` (Java Cryptographic Extension Keystore). Weitere Informationen hierzu finden Sie unter „Struktur der Keystore-Konfigurationsdatei (keystore.conf)” auf Seite 312.
- *%HOMEDRIVE%\%HOMEPATH%*`\.mqs\keystore.conf` (z. B. C:\Dokumente und Einstellungen\alice\.mqs\keystore.conf) ist die Standardposition, an der Erweiterte Nachrichtensicherheit nach der Datei `keystore.conf` sucht. Informationen zur Verwendung einer anderen Position als die Standardposition für die Datei `keystore.conf` finden Sie unter „Keystores und Zertifikate verwenden” auf Seite 311.
- Zum Erstellen eines `.mqs` -Verzeichnisses 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/alice/AMS/bobkey.kdb" -pw passw0rd
-label Bob_Cert -target bob_public.arm

runmqakm -cert -add -db "C:/Documents and Settings/bob/AMS/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

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 -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 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 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

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 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 weiteres 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. Als Benutzer Bob rufen Sie mithilfe einer Beispielanwendung eine Nachricht ab:

   ```
   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 abrufende Anwendung 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.

Schnelleinstieg für UNIX -Plattformen

Verwenden Sie dieses Handbuch, um IBM Erweiterte Nachrichtensicherheit schnell für die Nachrichtensicherheit auf UNIX -Plattformen zu konfigurieren. 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
- MQ Advanced Message Security

Die Komponentennamen auf den einzelnen Plattformen finden Sie in den folgenden Abschnitten:

- IBM WebSphere MQ-Komponenten für Linux-Systeme
- IBM WebSphere MQ-Komponenten für HP-UX-Systeme
- IBM WebSphere MQ-Komponenten für AIX-Systeme
- IBM WebSphere MQ-Komponenten für Solaris-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. Erweiterte Nachrichtensicherheit verwendet Interceptors zum Signieren und Verschlüsseln von Nachrichten an dem Punkt, an dem sie über die Standardschnittstelle von WebSphere MQ in die WebSphere MQ -Infrastruktur gelangen. Die Basiseinrichtung wird in WebSphere MQ vorgenommen und in den folgenden Schritten konfiguriert.

Sie können WebSphere 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 <MQ_INSTALL_PATH>/binverwenden. 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 müssen Sie es den beiden Benutzern ermöglichen, die Systemrichtlinienwarteschlange zu durchsuchen und Nachrichten in die Fehlerwarteschlange zu stellen.

   ```
   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
   ```

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 korrekt ausgeführt wurden, verwenden Sie die Beispiele amqsput und amqsget wie in Abschnitt „8. Verschlüsselung testen” auf Seite 300 beschrieben.

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 mithilfe von 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 „Schnelleinstieg für Java-Clients” auf Seite 300). 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

Sie müssen Erweiterte Nachrichtensicherheit -Interceptor 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 `.mqs` -Ordner 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 von `keystore.conf` wie folgt angezeigt:

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

Anmerkung:

- Der Pfad zu der Keystore-Datei muss ohne Dateierweiterung angegeben werden.
- Es gibt folgende Keystore-Formate: CMS (Syntax für verschlüsselte Nachrichten), JKS (Java-Keystore) und JCEKS (Java Cryptographic Extension Keystore). Weitere Informationen hierzu finden Sie unter „Struktur der Keystore-Konfigurationsdatei (keystore.conf)" auf Seite 312.
- `HOME/.mqs/keystore.conf` ist die Standardposition, an der Erweiterte Nachrichtensicherheit nach der Datei `keystore.conf` sucht. Informationen zur Verwendung einer anderen Position als die Standardposition für die Datei `keystore.conf` finden Sie unter „Keystores und Zertifikate verwenden" auf Seite 311.

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 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

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 abrufende Anwendung 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.

Schnelleinstieg für Java-Clients

Verwenden Sie dieses Handbuch, um IBM Erweiterte Nachrichtensicherheit schnell zu konfigurieren, um die Nachrichtensicherheit für Java-Anwendungen bereitzustellen, die über Clientbindungen eine Verbindung herstellen. Wenn Sie sie abgeschlossen haben, haben Sie einen Keystore erstellt, um Benutzeridentitäten zu überprüfen, und Sie haben für Ihren Warteschlangenmanager Richtlinien für Signatur/Verschlüsselung definiert.

Vorbereitende Schritte

Stellen Sie sicher, dass die entsprechenden Komponenten wie im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX) beschrieben installiert sind.

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. Erweiterte Nachrichtensicherheit verwendet Interceptors zum Signieren und Verschlüsseln von Nachrichten an dem Punkt, an dem sie über die Standardschnittstelle von WebSphere MQ in die WebSphere MQ -Infrastruktur gelangen. Die Basiseinrichtung wird in WebSphere 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 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 Szenario werden zwei Benutzer angezeigt: alice, der Absender, 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 wie im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX) 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
```

3. Außerdem müssen Sie es den beiden Benutzern ermöglichen, die Systemrichtlinienwarteschlange zu durchsuchen und Nachrichten in die Fehlerwarteschlange zu stellen.

```
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
```

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 Beispiele `JmsProducer` und `JmsConsumer` , wie im Abschnitt „7. Setup testen” auf Seite 305beschrieben.

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 eine Verbindung über Clientbindungen herstellen. Wenn Sie Java-Anwendungen mit lokalen Bindungen oder C-Anwendungen verwenden möchten, müssen Sie einen CMS-Keystore und Zertifikate mit dem Befehl **runmqakm** erstellen. Dies wird im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX) angezeigt.

Vorgehensweise

1. Erstellen Sie ein Verzeichnis, in dem Ihr Keystore erstellt werden soll, z. B. `/home/alice/.mqs`. Sie können es in demselben Verzeichnis erstellen, das auch im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX) 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 wird empfohlen, keine selbst

signierten Zertifikate zu verwenden, sondern stattdessen Zertifikate zu verwenden, 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 UNIX sicher, dass der Keystore lesbar ist.

```
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

Sie müssen Erweiterte Nachrichtensicherheit -Interceptor 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 verfügen. Dieser Schritt sollte sowohl für `alice` als auch für `bob` ausgeführt werden.

Beispiel

Für dieses Szenario lautet der Inhalt von `keystore.conf` for `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 lautet der Inhalt von `keystore.conf` for `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 ein `keystore.conf` haben, weil Sie dem **Leitfaden für den Schnelleinstieg** (Windows oder UNIX) gefolgt sind, können Sie das vorhandene bearbeiten, um es in den obigen Zeilen hinzuzufügen.
- Weitere Informationen finden Sie im Abschnitt „Struktur der Keystore-Konfigurationsdatei (keystore.conf)" auf Seite 312.

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 Tool IBM GSKit Keyman (ikeyman) unterscheidet beispielsweise zwischen dem *Extrahie-*

ren von Zertifikaten (öffentliche Schlüssel) und dem *Exportieren* privater Schlüssel. 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 WebSphere 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 WebSphere 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. Details zum Ausführen von Programmen unter verschiedenen Benutzern finden Sie im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX) für Ihre Plattform.

Vorgehensweise

1. Verwenden Sie zum Ausführen dieser JMS-Beispielanwendungen die Einstellung CLASSPATH für Ihre Plattform, wie in Von IBM WebSphere 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 abrufende Anwendung ausführt.

Ferne Warteschlangen schützen

Um ferne Warteschlangenverbindungen vollständig zu schützen, muss dieselbe Richtlinie in der fernen Warteschlange und in der lokalen Warteschlange, in die Nachrichten übertragen werden, definiert werden.

Wenn eine Nachricht in eine ferne Warteschlange gestellt wird, fängt Erweiterte Nachrichtensicherheit die Operation ab und verarbeitet die Nachricht entsprechend einer Maßnahmengruppe für die ferne Warteschlange. Für eine Verschlüsselungsrichtlinie wird die Nachricht z. B. verschlüsselt, bevor sie an den WebSphere MQ übergeben wird, um sie zu verarbeiten. Nachdem Erweiterte Nachrichtensicherheit die Nachricht verarbeitet hat, die in eine ferne Warteschlange gestellt wurde, stellt WebSphere MQ sie in die zugehörige Übertragungswarteschlange und leitet sie an den Zielwarteschlangenmanager und die Zielwarteschlange weiter.

Wenn eine GET-Operation für die lokale Warteschlange ausgeführt wird, versucht Erweiterte Nachrichtensicherheit , die Nachricht entsprechend dem Richtliniensatz in der lokalen Warteschlange zu dekodieren. 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 die Toleranzmarkierung, sobald das Erweiterte Nachrichtensicherheit -Rollout abgeschlossen ist.

Zugehörige Verweise

setmqspl (Sicherheitsrichtlinie festlegen)

Weiterleitung geschützter Nachrichten mithilfe von WebSphere Message Broker

IBM Erweiterte Nachrichtensicherheit kann Nachrichten in einer Infrastruktur schützen, in der WebSphere Message Broker Version 8.0.0.1 (oder höher) installiert ist. Sie sollten mit der Spezifik beider Produkte vertraut sein, bevor Sie die Sicherheit in der WebSphere Message Broker-Umgebung anwenden.

Informationen zu diesem Vorgang

Erweiterte Nachrichtensicherheit 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 zum Sichern von Nachrichten, die durch WebSphere Message Broker fließen, entweder WebSphere Message Broker erlauben können, Nachrichten zu verarbeiten, ohne ihren Inhalt zu kennen (Szenario 1), oder einen berechtigten Benutzer veranlassen können, Nachrichten zu empfangen und zu senden (Szenario 2).

Szenario 1-Nachrichtenbroker kann Nachrichteninhalt nicht anzeigen

Vorbereitende Schritte

Ihr WebSphere Message Broker 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 `routeTo`wird die Nachricht entweder an *bob* (`QBOB`) weitergeleitet.[1](`QCECIL`) oder die Standardwarteschlange (`QDEF`). Das Routing ist möglich, weil Erweiterte Nachrichtensicherheit nur die Nachrichtennutzdaten und nicht die zugehörigen Header und Eigenschaften schützt, die ungeschützt bleiben und von WebSphere Message Broker gelesen werden können. Erweiterte Nachrichtensicherheit wird nur von *alice*, *bob* und *cecil*verwendet. Es ist nicht erforderlich, ihn für WebSphere Message Broker zu installieren oder zu konfigurieren.

WebSphere Message Broker empfängt die geschützte Nachricht aus der ungeschützten Aliaswarteschlange, um jeden Versuch zu vermeiden, die Nachricht zu entschlüsseln. 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 von WebSphere Message Broker weitergeleitet und kommt unverändert in der Zielwarteschlange an. Daher wird sie weiterhin vom ursprünglichen Autor signiert (*bob* und *cecil* akzeptieren nur Nachrichten, die von *alice*gesendet wurden) und wie zuvor geschützt (nur *bob* und *cecil* können sie lesen. WebSphere Message Broker reiht die weitergeleitete Nachricht in einen ungeschützten Aliasnamen ein. Die Empfänger rufen die Nachricht aus einer geschützten Ausgabewarteschlange ab, in der die Nachricht von IBM WebSphere MQ AMS transparent entschlüsselt wird.

[1] Cecil's

Vorgehensweise

1. Konfigurieren Sie *alice*, *bob* und *cecil* zur Verwendung von Erweiterte Nachrichtensicherheit, wie im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX) 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 beschriebene Methode verwenden. Gemeinsame Nutzung von Zertifikaten** im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX).
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 Warteschlange QIN entsprechend einer der infrage kommenden Konfigurationen 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.

   ```
   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 WebSphere Message Broker einen Nachrichtenfluss, um die Nachrichten, die in der AIN -Aliaswarteschlange ankommen, abhängig von der Eigenschaft routeTo der Nachricht an den BOB-, CECIL-oder DEF-Knoten weiterzuleiten. 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 den 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 WebSphere Message Broker-Laufzeitkomponente.
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 vom WebSphere Message Broker aus der Aliaswarteschlange `AIN` abgerufen. WebSphere Message Broker entscheidet, wohin die Nachricht, die die Eigenschaft `routeTo` liest, weitergeleitet werden soll, da alle Eigenschaften nicht verschlüsselt sind. WebSphere Message Broker stellt die Nachricht in den entsprechenden ungeschützten Aliasnamen und verhindert so den weiteren Schutz. Wird die Nachricht von *bob* oder *cecil* aus der Warteschlange empfangen, wird die Nachricht entschlüsselt und die digitale Signatur geprüft.

Szenario 2-Nachrichtenbroker kann Nachrichteninhalt anzeigen

Informationen zu diesem Vorgang

In diesem Szenario darf eine Gruppe von Einzelpersonen Nachrichten an WebSphere Message Broker senden. Eine andere Gruppe ist berechtigt, Nachrichten zu empfangen, die von WebSphere Message Broker erstellt wurden. Die Übertragung zwischen den Parteien und WebSphere Message Broker kann nicht abgehört werden.

Denken Sie daran, dass WebSphere Message Broker Schutzrichtlinien und Zertifikate nur liest, wenn eine Warteschlange geöffnet wird. Daher müssen Sie die Ausführungsgruppe erneut laden, nachdem Sie alle Änderungen an Schutzrichtlinien vorgenommen haben, damit die Änderungen wirksam werden.

```
mqsireload execution-group-name
```

Wenn WebSphere Message Broker als berechtigte Partei betrachtet wird, die die Nachrichtennutzdaten lesen oder signieren darf, müssen Sie Erweiterte Nachrichtensicherheit für den Benutzer konfigurieren, der den WebSphere Message Broker-Service startet. Beachten Sie, dass es nicht unbedingt derselbe Benutzer ist, der die Nachrichten in Warteschlangen einreiht/abruft, noch der Benutzer, der die WebSphere Message Broker-Anwendungen erstellt und implementiert.

Vorgehensweise

1. Konfigurieren Sie *alice*, *bob*, *cecil* und *dave* sowie den WebSphere Message Broker-Servicebenutzer für die Verwendung von Erweiterte Nachrichtensicherheit gemäß der Beschreibung im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX).

   Stellen Sie sicher, dass die folgenden Schritte ausgeführt werden:

   - Benutzer erstellen und berechtigen

- Schlüsseldatenbank und Zertifikate erstellen
- Keystore.conf wird erstellt

2. Stellen Sie *alice*-, *bob*-, *cecil* -und *dave' s* -Zertifikate für den Benutzer des WebSphere Message Broker-Service bereit.

   Extrahieren Sie dazu alle Zertifikate, die *alice*, *bob*, *cecil* und *dave*identifizieren, in externe Dateien und fügen Sie anschließend die extrahierten Zertifikate zum WebSphere Message Broker-Keystore hinzu. Es ist wichtig, dass Sie die in **Aufgabe 5 beschriebene Methode verwenden. Gemeinsame Nutzung von Zertifikaten** im **Leitfaden für den Schnelleinstieg** (Windows oder UNIX).
3. Stellen Sie das WebSphere Message Broker Service-Benutzerzertifikat für *alice*, *bob*, *cecil* und *dave*bereit.

   Verwenden Sie dazu die im vorherigen Schritt angegebene Methode.

   Anmerkung: *Alice* und *bob* benötigen das Benutzerzertifikat des WebSphere Message Broker Service, um die Nachrichten ordnungsgemäß zu verschlüsseln. Der Benutzer des WebSphere Message Broker-Service benötigt Zertifikate von *alice* und *bob* , um die Autoren der Nachrichten zu überprüfen. Der WebSphere Message Broker-Servicebenutzer benötigt *cecil-Zertifikate* und *dave' s* , um die Nachrichten für sie zu verschlüsseln. *cecil* und *dave* benötigen das Zertifikat des Benutzers von WebSphere Message Broker Service, um zu überprüfen, ob die Nachricht von WebSphere Message Broker stammt.
4. Definieren Sie eine lokale Warteschlange mit dem Namen `IN` und definieren Sie die Sicherheitsrichtlinie mit *alice* und *bob* als Autoren und WebSphere Message Broker-Servicebenutzer als Empfänger:

   ```
   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 dem Namen OUT und definieren Sie die Sicherheitsrichtlinie mit dem WebSphere Message Broker-Servicebenutzer, der als Autor und *cecil* und *dave* als Empfänger angegeben ist:

   ```
   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 WebSphere Message Broker einen Nachrichtenfluss mit einem `MQInput` -und `MQOutput` -Knoten. 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 WebSphere Message Broker-Laufzeitkomponente.
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, werden verschlüsselt, sodass nur WebSphere Message Broker sie lesen kann. WebSphere Message Broker akzeptiert nur Nachrichten von *alice* und *bob* und weist alle anderen zurück. Die akzeptierten Nachrichten werden entsprechend verarbeitet, dann signiert und mit *cecils* und *dave' s* -Schlüsseln verschlüsselt, bevor sie in die Ausgabewarteschlange OUTeingereiht werden. Nur *cecil* und *dave* können ihn lesen, Nachrichten, die nicht von WebSphere Message Broker signiert wurden, werden zurückgewiesen.

IBM WebSphere MQ Advanced Message Security mit IBM WebSphere MQ Managed File Transfer verwenden

In diesem Szenario wird erläutert, wie Erweiterte Nachrichtensicherheit konfiguriert wird, um die Vertraulichkeit von Daten für Daten bereitzustellen, die über eine IBM WebSphere MQ Managed File Transfer gesendet werden.

Vorbereitende Schritte

Stellen Sie sicher, dass die Erweiterte Nachrichtensicherheit -Komponente in der WebSphere MQ -Installation installiert ist, in der sich die von IBM WebSphere MQ Managed File Transfer verwendeten Warteschlangen befinden, die geschützt werden sollen.

Wenn Ihre IBM WebSphere MQ Managed File Transfer -Agenten eine Verbindung im Bindungsmodus herstellen, stellen Sie sicher, dass Sie auch die GSKit-Komponente in ihrer lokalen Installation installiert haben.

Informationen zu diesem Vorgang

Wenn die Übertragung von Daten zwischen zwei IBM WebSphere MQ Managed File Transfer -Agenten unterbrochen wird, bleiben möglicherweise vertrauliche Daten in den zu Grunde liegenden WebSphere MQ -Warteschlangen, die zur Verwaltung der Übertragung verwendet werden, ungeschützt. In diesem Szenario wird beschrieben, wie Sie Erweiterte Nachrichtensicherheit konfigurieren und verwenden, damit solche Daten in den IBM WebSphere MQ Managed File Transfer-Warteschlangen geschützt sind.

In diesem Szenario wird eine einfache Topologie betrachtet, die aus einer Maschine mit zwei IBM WebSphere MQ Managed File Transfer -Warteschlangen und zwei Agenten (AGENT1 und AGENT2) besteht und einen einzelnen Warteschlangenmanager (hubQM) gemeinsam nutzt, wie im Szenario Basisdateiübertragung unter Verwendung der Scriptsbeschrieben. Beide Agenten verbinden sich auf die gleiche Weise, entweder im Bindungsmodus oder im Clientmodus.

1. Zertifikate erstellen

Vorbereitende Schritte

Dieses Szenario zeigt ein einfaches Modell, in dem Benutzer `ftagent` der Gruppe FTAGENTS zur Ausführung der Agentenprozesse für IBM WebSphere MQ Managed File Transfer verwendet wird. Wenn Sie Ihre eigenen Benutzer-und Gruppennamen verwenden, ändern Sie die Befehle entsprechend.

Informationen zu diesem Vorgang

Erweiterte Nachrichtensicherheit verwendet die Verschlüsselung mit öffentlichen Schlüsseln, um Nachrichten in geschützten Warteschlange zu signieren und/oder zu verschlüsseln.

Anmerkung:

- Wenn Ihre IBM WebSphere MQ Managed File Transfer -Agenten im Bindungsmodus ausgeführt werden, werden die Befehle, die Sie zum Erstellen eines CMS-Keystores (CMS = Cryptographic Message Syntax) verwenden, im **Leitfaden für den Schnelleinstieg** (Fenster oder UNIX) für Ihre Plattform ausführlich beschrieben.
- Wenn Ihre IBM WebSphere MQ Managed File Transfer -Agenten im Clientmodus ausgeführt werden, werden die Befehle, die Sie zum Erstellen eines JKS (Java Keystore) benötigen, im Abschnitt „Schnelleinstieg für Java-Clients” auf Seite 300beschrieben.

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=IBM, L=Hursley, ST=Hampshire, C=GB
   ```

2. Erstellen Sie eine `keystore.conf`-Datei, um die Position des Keystores und des Zertifikats in der Datei anzugeben, wie im entsprechenden 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 IBM WebSphere MQ 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=IBM, L=Hurs
ley, ST=Hampshire, C=GB"
-e AES128 -r "CN=ftagent, OU=MFT, O=IBM, L=Hursley, ST=Hampshire, C=GB"
```

3. Stellen Sie sicher, dass der Benutzer, der den IBM WebSphere MQ Managed File Transfer-Agentenprozess ausführt, über die Berechtigung zum Durchsuchen der Systemrichtlinienwarteschlange sowie zum Einreihen von Nachrichten in die Fehlerwarteschlange verfügt.

```
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 IBM WebSphere MQ 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.

IBM WebSphere MQ Advanced Message Security installieren

Sie können die IBM WebSphere MQ Advanced Message Security-Komponente auf verschiedenen Plattformen installieren.

Informationen zu diesem Vorgang

Das vollständige Installationsverfahren finden Sie unter IBM WebSphere MQ Advanced Message Security installieren.

Zugehörige Tasks

DeinstallationIBM WebSphere MQ Advanced Message Security

Keystores und Zertifikate verwenden

Um für WebSphere MQ -Anwendungen einen transparenten Verschlüsselungsschutz bereitzustellen, verwendet Erweiterte Nachrichtensicherheit die Schlüsselspeicherdatei, in der öffentliche Schlüsselzertifikate und ein privater Schlüssel gespeichert werden.

In Erweiterte Nachrichtensicherheit 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.

Die Position des Keystores wird in der Keystore-Konfigurationsdatei bereitgestellt, die standardmäßig `keystore.conf` ist. Jeder Erweiterte Nachrichtensicherheit -Benutzer muss über die Keystore-Konfigurationsdatei verfügen, die auf eine Keystore-Datei zeigt. Erweiterte Nachrichtensicherheit akzeptiert das folgende Format von Keystore-Dateien: `.kdb`, `.jceks`, `.jks`.

Die Standardposition der Datei `keystore.conf` lautet wie folgt:

- Auf UNIX -Plattformen: `$HOME/.mqs/keystore.conf`
- Auf Windows-Plattformen: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

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 UNIX and Linux: `export MQS_KEYSTORE_CONF=`*path/filename*
  - Unter Windows: `set MQS_KEYSTORE_CONF=`*path\filename*

    Anmerkung: Der Pfad aufWindows kann und sollte den Laufwerksbuchstaben angeben, wenn mehr als ein Laufwerksbuchstabe vorhanden ist.

Zugehörige Konzepte

„Distinguished Names für Absender” auf Seite 326
Die definierten Namen (DNs) des Absenders identifizieren Benutzer, die berechtigt sind, Nachrichten in eine Warteschlange einzureihen.

„Distinguished Names für Empfänger” auf Seite 327
Die definierten Namen (DN) des Empfängers geben Benutzer an, die berechtigt sind, Nachrichten aus einer Warteschlange abzurufen.

Struktur der Keystore-Konfigurationsdatei (keystore.conf)

Die Schlüsselspeicherkonfigurationsdatei (`keystore.conf`) verweist Erweiterte Nachrichtensicherheit auf die Position des entsprechenden Schlüsselspeichers.

Es gibt zwei Typen von Konfigurations-CMS und Java (JKS und JCEKS). CMS-Konfigurationseinträge haben das Präfix `cms.` und Java das Präfix `jks.` oder `jceks.` , je nach Typ eines Keystores.

Die Konfigurationsdatei kann je nach Typ der Konfigurationsdatei eine der folgenden Strukturen aufweisen:

```
cms.keystore = /<dir>/<keystore_file>
cms.certificate = certificate_label

jceks.keystore = <dir>/Keystore
jceks.certificate = <certificate_label>
jceks.encrypted = no
jceks.keystore_pass = <password>
jceks.key_pass = <password>
jceks.provider = IBMJCE

jks.keystore = <dir>/Keystore
jks.certificate = <certificate_label>
jks.encrypted = no
jks.keystore_pass = <password>
jks.key_pass = <password>
jks.provider = IBMJCE
```

Konfigurationsdateiparameter werden wie folgt definiert:

keystore
Pfad zur Keystore-Datei.

Wichtig:

- Der Pfad zu der Keystore-Datei darf die Dateierweiterung nicht enthalten.

- Für Java-Keystore-Dateien unterstützt IBM WebSphere MQ AMS die folgenden Dateiformate: `.jks`, `.jceks`, `.jck`.

`certificate`
Zertifikatsbezeichnung.

`encrypted`
Status des Kennworts.

`keystore_pass`
Kennwort für die Schlüsselspeicherdatei.

Anmerkung:

- Für den CMS-Keystore stützt sich IBM WebSphere MQ AMS auf die Stashdateien (`.sth`), während JKS und JCEKS unter Umständen ein Kennwort für das Zertifikat und den privaten Schlüssel des Benutzers benötigen.
- Das Speichern von Kennwörtern in Klartext stellt ein Sicherheitsrisiko dar.

`key_pass`
Kennwort für den privaten Schlüssel des Benutzers.

Wichtig: Das Speichern von Kennwörtern in Klartextform kann ein Sicherheitsrisiko darstellen.

`provider`
Der Java-Sicherheitsprovider, der Verschlüsselungsalgorithmen implementiert, die für das Keystore-Zertifikat erforderlich sind

Anmerkung: Derzeit ist IBMJCE der einzige Provider, der von Erweiterte Nachrichtensicherheitunterstützt wird.

Wichtig: Die im Keystore gespeicherten Informationen sind für den sicheren Datenfluss, der mithilfe von WebSphere MQgesendet wird, von entscheidender Bedeutung. Aus diesem Grund müssen Sicherheitsadministratoren bei der Zuordnung von Dateiberechtigungen zu diesen Dateien besondere Aufmerksamkeit schenken.

Es folgt ein Beispiel für die Datei `keystore.conf` :

```
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = <password>
jceks.key_pass = <password>
jceks.provider = IBMJCE
```

Zugehörige Tasks

„Kennwörter in Java schützen” auf Seite 323
Das Speichern von Schlüsselwörtern für Schlüsselspeicher und private Schlüssel als einfacher Text stellt ein Sicherheitsrisiko dar, sodass Erweiterte Nachrichtensicherheit ein Tool zur Verfügung stellt, das diese Kennwörter mit Hilfe eines Benutzerschlüssels, der in der Schlüsselspeicherdatei verfügbar ist, verwürfelt.

Überwachung des Nachrichtenkanalagenten (MCA)

Das MCA-Abfangen ermöglicht es einem Warteschlangenmanager, der unter IBM WebSphere MQ ausgeführt wird, Richtlinien selektiv für Serververbindungskanäle anzuwenden.

Durch MCA-Abfangen können Clients, die außerhalb von IBM WebSphere MQ AMS bleiben, immer noch mit einem Warteschlangenmanager verbunden sein und ihre Nachrichten verschlüsselt und entschlüsselt werden.

Das MCA-Abfangen soll die IBM WebSphere MQ AMS -Funktion bereitstellen, wenn IBM WebSphere MQ AMS nicht auf dem Client aktiviert werden kann. Beachten Sie, dass die Verwendung des MCA-Abfangens

und eines IBM WebSphere MQ AMS -fähigen Clients zu einem doppelten Schutz von Nachrichten führt, die für den Empfang von Anwendungen problematisch sein könnten.

Wenn ein Fehler `2085 (MQRC_UNKNOWN_OBJECT_NAME)` gemeldet wird, wenn Sie einen Version 7.5 -Client oder höher verwenden, um eine Verbindung zu einem WS-Manager aus einer früheren Version des Produkts herzustellen, müssen Sie IBM WebSphere MQ Advanced Message Security auf dem Client inaktivieren. Weitere Informationen finden Sie unter „IBM WebSphere MQ Advanced Message Security auf dem Client inaktivieren” auf Seite 316.

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 Keystore von IBM WebSphere MQ AMS finden Sie im Abschnitt „Keystores und Zertifikate verwenden” auf Seite 311.

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.

Ein Beispiel für die Einrichtung des MCA-Abfangens finden Sie unter „IBM WebSphere MQ AMS Überwachungsbeispiel für MCA” auf Seite 314.

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 WebSphere MQ AMS Überwachungsbeispiel für MCA

Eine Beispieltask für das Einrichten der Abfangfunktion mit dem IBM WebSphere MQ AMS MCA.

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.

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 Version 7.5 stellte IBM WebSphere MQ AMS ein Add-on-Produkt dar, das separat installiert werden musste und bei dem Abfangprozesse zum Schutz von Anwendungen konfiguriert werden mussten. Seit Version 7.5 sind die Interceptors automatisch integriert und werden im MQ-Client und den Serverlaufzeitumgebungen 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 der Version 7.5 oder höher verwendet wird, wird die Nachricht doppelt geschützt, da die Nachricht vom Interceptor für die MQ-Clientlaufzeit und vom MCA-Interceptor beim Empfang in MQ geschützt wird.

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/mq75
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 UNIX).
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 Version 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 WebSphere MQ Version 7.5-Installation:

```
/opt/mq75/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 WebSphere MQ Version 7.5-Installation aus:

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

Zugehörige Tasks

„Schnelleinstieg für Java-Clients" auf Seite 300
Verwenden Sie dieses Handbuch, um IBM Erweiterte Nachrichtensicherheit schnell zu konfigurieren, um die Nachrichtensicherheit für Java-Anwendungen bereitzustellen, die über Clientbindungen eine Verbindung herstellen. Wenn Sie sie abgeschlossen haben, haben Sie einen Keystore erstellt, um Benutzeridentitäten zu überprüfen, und Sie haben für Ihren Warteschlangenmanager Richtlinien für Signatur/Verschlüsselung definiert.

Zugehörige Verweise

„Bekannte Einschränkungen" auf Seite 288
Informationen zu Einschränkungen von IBM WebSphere MQ Advanced Message Security.

IBM WebSphere MQ Advanced Message Security auf dem Client inaktivieren

Sie müssen IBM WebSphere MQ Advanced Message Security (AMS) auf dem Client inaktivieren, wenn Sie einen Client mit Version 7.5 oder höher verwenden, um eine Verbindung zu einem Warteschlangenmanager einer früheren Version des Produkts herzustellen, und ein Fehler 2085 (MQRC_UNKNOWN_OBJECT_NAME) gemeldet wird.

Informationen zu diesem Vorgang

Ab Version 7.5ist IBM WebSphere MQ Advanced Message Security (AMS) automatisch in einem IBM WebSphere MQ -Client aktiviert, sodass der Client standardmäßig versucht, die Sicherheitsrichtlinien für Objekte im Warteschlangenmanager zu überprüfen. Auf Servern mit früheren Versionen des Produkts, z. B. Version 7.1, ist AMS jedoch nicht aktiviert, was dazu führt, dass ein Fehler 2085 (MQRC_UNKNOWN_OBJECT_NAME) gemeldet wird.

Wenn dieser Fehler gemeldet wird, wenn Sie versuchen, eine Verbindung zu einem Warteschlangenmanager einer früheren Version des Produkts herzustellen, können Sie AMS auf dem Client wie folgt inaktivieren:

- Für Java-Clients haben Sie folgende Möglichkeiten:
  - V 7.5.0.4 Durch Festlegen einer Umgebungsvariablen AMQ_DISABLE_CLIENT_AMS.
  - V 7.5.0.4 Durch Festlegen der Java-Systemeigenschaft com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS.
  - V 7.5.0.5 Durch Verwendung der Eigenschaft DisableClientAMS in der Zeilengruppe **Security** in der Datei `mqclient.ini` .
- Für C-Clients auf eine der folgenden Arten:
  - V 7.5.0.4 Durch Festlegen einer Umgebungsvariablen AMQ_DISABLE_CLIENT_AMS.
  - V 7.5.0.5 Durch Verwendung der Eigenschaft DisableClientAMS in der Zeilengruppe **Security** in der Datei `mqclient.ini` .

Prozedur

- Verwenden Sie eine der folgenden Optionen, um AMS auf dem Client zu inaktivieren:

V 7.5.0.4 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 Version 7.5oder höher verwenden, IBM WebSphere MQ classes for Java -oder IBM WebSphere MQ classes for JMS -Client.

Sie können auch AMQ_DISABLE_CLIENT_AMS verwenden, um die AMS -Funktion für C-Clients zu inaktivieren.

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
```

V 7.5.0.4 Systemeigenschaft com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS

Setzen Sie für IBM WebSphere MQ classes for JMS -und IBM WebSphere MQ classes for Java -Clients die Java -Systemeigenschaft com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS für die Java -Anwendung auf den Wert TRUE .

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.

V 7.5.0.5 `DisableClientAMS`-Eigenschaft in der `mqclient.ini`-Datei

Fügen Sie für IBM WebSphere MQ classes for JMS -und IBM WebSphere MQ classes for Java -Clients und für C-Clients den Eigenschaftsnamen `DisableClientAMS` unter der Zeilengruppe **`Security`** der Datei `mqclient.ini` hinzu, wie im folgenden Beispiel gezeigt:

```
Security:
DisableClientAMS=Yes
```

Sie können AMS auch aktivieren, wie im folgenden Beispiel gezeigt:

```
Security:
DisableClientAMS=No
```

Nächste Schritte

Weitere Informationen zu Problemen beim Öffnen von Warteschlangen, die durch AMS geschützt werden, finden Sie im Abschnitt „Probleme beim Öffnen geschützter Warteschlangen bei der Verwendung von JMS” auf Seite 342.

Zugehörige Konzepte

„Überwachung des Nachrichtenkanalagenten (MCA)” auf Seite 313
Das MCA-Abfangen ermöglicht es einem Warteschlangenmanager, der unter IBM WebSphere MQ ausgeführt wird, Richtlinien selektiv für Serververbindungskanäle anzuwenden.

Zugehörige Tasks

Client mit einer Konfigurationsdatei konfigurieren

Zugehörige Verweise

Die Konfigurationsdatei IBM WebSphere MQ classes for JMS

Zertifikatsanforderungen für AMS

Zertifikate müssen über einen öffentlichen RSA-Schlüssel verfügen, damit sie mit Erweiterte Nachrichtensicherheit verwendet werden können.

Weitere Informationen zu verschiedenen Typen öffentlicher Schlüssel und deren Erstellung finden Sie unter „Digitale Zertifikate und CipherSpec -Kompatibilität in IBM WebSphere MQ" auf Seite 36.

Schlüsselverwendungserweiterungen

Die Schlüsselverwendungserweiterungen stellen zusätzliche Einschränkungen für die Verwendung eines Zertifikats dar.

In Erweiterte Nachrichtensicherheit muss die Schlüsselnutzung folgendermaßen festgelegt werden: Für Zertifikate in X.509 V3 oder höher mit dem Datenschutzniveau 'Integrity', müssen sie, falls die Erweiterungen der Schlüsselnutzung festgelegt sind, mindestens eine der beiden folgenden Erweiterungen enthalten:

- **nonRepudiation**
- **digitalSignature**

Wenn die Schlüsselverwendungserweiterungen festgelegt sind, müssen Sie für die Datenschutzqualität auch die Erweiterung **keyEncipherment** einschließen.

Zugehörige Konzepte

„Qualität des Schutzes" auf Seite 329
Erweiterte Nachrichtensicherheit-Richtlinien für den Datenschutz beinhalten ein Datenschutzniveau (Quality of Protection, QOP).

Methoden zur Zertifikatsprüfung in IBM WebSphere MQ Advanced Message Security

Sie können mit IBM WebSphere MQ 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 IBM WebSphere MQ AMS können Sie die Zertifikatsgültigkeit mit Online Certificate Status Protocol (OCSP) oder mit der Zertifikatsperrliste (CRL) prüfen.

IBM WebSphere MQ AMS kann für die Prüfung von OCSP und/oder der CRL konfiguriert werden. Wenn beide Methoden aktiviert sind, verwendet IBM WebSphere MQ AMS aus Leistungsgründen zuerst OCSP für den Widerrufsstatus. Wenn der Widerrufsstatus eines Zertifikats nach der OCSP-Prüfung nicht ermittelt werden kann, verwendet IBM WebSphere MQ AMS die CRL-Prüfung.

Zugehörige Konzepte

„OCSP (Online Certificate Status Protocol)" auf Seite 318
OCSP (Online Certificate Status Protocol) bestimmt, ob ein Zertifikat widerrufen wurde, und hilft daher festzustellen, ob das Zertifikat vertrauenswürdig ist.

„Zertifikatswiderrufslisten (CRLs)" auf Seite 320
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)

OCSP (Online Certificate Status Protocol) bestimmt, ob ein Zertifikat widerrufen wurde, und hilft daher festzustellen, ob das Zertifikat vertrauenswürdig ist.

OCSP-Prüfung in nativen Abfangprozessen aktivieren

Zu Aktivierung der OCPS-Prüfung (Online Certificate Status Protocol) in Erweiterte Nachrichtensicherheit müssen Sie die Schlüsselspeicherkonfigurationsdatei ändern.

Vorgehensweise

Fügen Sie der Schlüsselspeicherkonfigurationsdatei die folgenden Optionen hinzu:

Anmerkung: Bei den in der Tabelle angegebenen Werte für einzelne Optionen handelt es sich um Standardwerte.

Sie müssen einen der folgenden Werte angeben:

- `ocsp.enable=on`
- `ocsp.url=<reposnder_URL>`
- `ocsp.http.proxy.host=<OCSP_proxy>`

Option	Beschreibung
`ocsp.enable=off`	Aktivieren Sie die OCSP-Prüfung, wenn das Zertifikat, das geprüft wird, über eine Authority Info Access-Erweiterung mit der Zugriffsmethode PKIX_AD_OCSP verfügt, die einen URI für die Position des OCSP-Responders enthält. Mögliche Wert sind: `on`/`off`.
`ocsp.url=<reposnder_URL>`	Die URL-Adresse des OCSP-Responder.
`ocsp.http.proxy.host=<OCSP_proxy>`	Die URL-Adresse des OCSP-Proxy-Servers.
`ocsp.http.proxy.port=<port_number>`	Die Port-Nummer des OCSP-Proxy-Servers.
`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.
`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 Erweiterte Nachrichtensicherheit überschritten ist.

OCSP-Prüfung in Java aktivieren
Um das OCSP-Check-in für Java in Erweiterte Nachrichtensicherheitzu aktivieren, ändern Sie die Datei `java.security` oder die Keystore-Konfigurationsdatei.

Informationen zu diesem Vorgang

Es gibt zwei Möglichkeiten, die OCSP-Prüfung in Erweiterte Nachrichtensicherheit zu aktivieren:

Verwenden von ' java.security
Überprüfen Sie, ob für Ihr Zertifikat AIA (Authority Information Access) konfiguriert ist.

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 Java Security Manager verwenden, fügen Sie auch die folgende Java-Berechtigung zu `lib/security/java.policy` hinzu, um die Konfiguration abzuschließen

```
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 die folgenden Java-Berechtigungen zu `lib/security/java.policy`hinzu, um die Konfiguration abzuschließen:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

Zertifikatswiderrufslisten (CRLs)

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 Validieren von Zertifikaten erstellt Erweiterte Nachrichtensicherheit eine Zertifikatskette, die aus dem Zertifikat des Unterzeichners und der Zertifikatskette 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.IBM WebSphere MQ AMS überprüft den Zertifikatspfad mit einem PKIX-Validierungsalgorithmus. Wenn die Kette erstellt und überprüft ist, schließt IBM WebSphere MQ 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, prüft IBM WebSphere MQ 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 IBM WebSphere MQ AMS CRLs von Dateien oder von LDAP herunter, abhängig von den Werten, die in der Konfigurationsdatei angegeben wurden. Nur CRLs, die im DER-Format codiert sind, werden von IBM WebSphere MQ AMS unterstützt.

Wenn sich in der Konfigurationsdatei des Keystores keine CRL-bezogene Konfiguration befindet, führt IBM WebSphere MQ AMS keine Gültigkeitsprüfung für CRLs aus. Für jedes CA-Zertifikat fragt IBM WebSphere MQ AMS LDAP für CRLs unter Verwendung von Distinguished Names einer CA ab, um die 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 des Schlüsselspeichers so ändern, dass Erweiterte Nachrichtensicherheit CLRs vom LDAP-Server (Lightweight Directory Access Protocol) herunterladen kann.

Vorgehensweise

Fügen Sie der Konfigurationsdatei die folgenden Optionen hinzu:

Anmerkung: Bei den in der Tabelle angegebenen Werte für einzelne Optionen handelt es sich um Standardwerte.

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 Java-Interceptor von IBM WebSphere MQ AMS erfolgreich eine Verbindung zu einem LDAP-Server herstellt, 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 beim LDAP-Server an. Wenn dieser Wert nicht angegeben wird, müssen die CRL-Attribute in LDAP weltlesbar sein.
`crl.ldap.pass=<`*`password`*`>`	Kennwort für den LDAP-Server.
`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.

Option	Beschreibung
`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`	Wartezeit auf eine Serverantwort in Sekunden, nach deren Ablauf IBM WebSphere MQ AMS das Zeitlimit überschritten hat.
`crl.http.cache_size=0`	HTTP-Cachegröße in Byte.

Unterstützung von Zertifikatswiderrufslisten in Java aktivieren
Um die CRL-Unterstützung in Erweiterte Nachrichtensicherheit zu aktivieren, müssen Sie die Keystore-Konfigurationsdatei ändern, damit IBM WebSphere MQ AMS CRLs vom 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.
`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 Java-Interceptor von IBM WebSphere MQ AMS erfolgreich eine Verbindung zu einem LDAP-Server herstellt, versucht er nicht, CRLs von den übrigen bereitgestellten Servern herunterzuladen. Java verwendet keine Werte für `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 Datei JRE/`lib/security/java.security` 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.
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.

Kennwörter in Java schützen

Das Speichern von Schlüsselwörtern für Schlüsselspeicher und private Schlüssel als einfacher Text stellt ein Sicherheitsrisiko dar, sodass Erweiterte Nachrichtensicherheit ein Tool zur Verfügung stellt, das diese Kennwörter mit Hilfe eines Benutzerschlüssels, der in der Schlüsselspeicherdatei verfügbar ist, verwürfelt.

Vorbereitende Schritte

Der keystore.conf -Dateieigner muss sicherstellen, dass nur der Dateieigner berechtigt ist, die Datei zu lesen. Der in diesem Kapitel beschriebene Schutz von Kennwörtern ist nur ein zusätzliches Maß an Schutz.

Vorgehensweise

1. Bearbeiten Sie die keystore.conf -Dateien, um den Pfad zum Keystore und zu den Benutzerbeschriften einzuschließen.

```
jceks.keystore = c:/Documents and Setting/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.provider = IBMJCE
```

2. Geben Sie Folgendes ein, um das Tool auszuführen:

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password
```

Eine Ausgabe mit verschlüsselten Kennwörtern wird generiert und kann in die `keystore.conf` -Datei kopiert werden.

Führen Sie folgenden Befehl aus, um die Ausgabe automatisch in die Datei `keystore.conf` zu kopieren:

```
java -cp com.ibm.mq.jmqi.jar com.ibm.mq.ese.config.KeyStoreConfigProtector keystore_password
private_key_password >> ~/<path_to_keystore>/keystore.conf
```

Anmerkung:

Eine Liste der Standardpositionen von `keystore.conf` auf verschiedenen Plattformen finden Sie in „Keystores und Zertifikate verwenden” auf Seite 311.

Beispiel

Es folgt ein Beispiel für eine solche Ausgabe:

```
#Fri Jul 30 15:20:29 CEST 2010
jceks.key_pass=MMXh997n5ZOr8uRlJmc5qity9MN2CggGBMKCDxdbn1AyPklvdgTsOLG6X3C1YT7oD
zwaqZFlOR4t\r\nmZsc7JGAx8nqqxLnAucdGn0NWo6xnjZB1n501YGol2k/PhaQHhFXKMAU9dKg0f8djOtCA
OlX4ETe\r\nfYl9LBUt2wk87uM7dSs\=
jceks.keystore_pass=OIdeayBnSCfLG4cFuxEVrk6SYyAsdSPpDqgPf16s9s1M04cqZjNbhgjoA2EXo
nudHZHH+4s2drvQ\r\nCUvQgu9GuaBMJK2F2OjtHJJ1Y4BVeLW2c2okgawo/W2J1AdUYKkJ0raYTkDouLaTYTQeul
yG0xIl\r\niD2si1xUCxhYvvyhbbY\=
jceks.encrypted=yes
```

IBM WebSphere MQ Advanced Message Security-Sicherheitsrichtlinien anwenden

IBM WebSphere MQ 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 Sicherheitsrichtlinien

IBM Erweiterte Nachrichtensicherheit -Sicherheitsrichtlinien sind konzeptionelle Objekte, die beschreiben, wie eine Nachricht kryptografisch verschlüsselt und signiert wird.

Ausführliche Informationen zu den Attributen der Sicherheitsrichtlinie finden Sie in den folgenden Unterabschnitten:

Zugehörige Konzepte

„Qualität des Schutzes” auf Seite 329
Erweiterte Nachrichtensicherheit-Richtlinien für den Datenschutz beinhalten ein Datenschutzniveau (Quality of Protection, QOP).

„Sicherheitsrichtlinienattribute” auf Seite 328
Sie können Erweiterte Nachrichtensicherheit verwenden, um einen bestimmten Algorithmus oder eine bestimmte Methode zum Schutz der Daten auszuwählen.

Richtlinienname

Der Richtlinienname ist ein eindeutiger Name, der eine bestimmte Erweiterte Nachrichtensicherheit -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 Erweiterte Nachrichtensicherheit -Richtlinie (IBM WebSphere MQ 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 IBM WebSphere MQ 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

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

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 Erweiterte Nachrichtensicherheit -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.

Tolerierung

Das Toleranzattribut gibt an, ob IBM Erweiterte Nachrichtensicherheit Nachrichten ohne angegebene Sicherheitsrichtlinie akzeptieren kann.

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 (**default**)

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.

Distinguished Names für Absender

Die definierten Namen (DNs) des Absenders identifizieren Benutzer, die berechtigt sind, Nachrichten in eine Warteschlange einzureihen.

IBM Erweiterte Nachrichtensicherheit (IBM WebSphere MQ AMS) überprüft nicht, ob eine Nachricht von einem gültigen Benutzer in eine datengeschützte Warteschlange eingereiht wurde, bis die Nachricht abgerufen wurde. Wenn die Richtlinie einen oder mehrere gültige Absender festlegt und der Benutzer, der die Nachricht in die Warteschlange gestellt hat, sich nicht in der Liste der gültigen Absender befindet, gibt IBM WebSphere MQ AMS zu diesem Zeitpunkt einen Fehler an die abrufende Anwendung zurück und stellt die Nachricht in ihre Fehlerwarteschlange.

Eine Richtlinie kann 0 oder mehr Absender-DNs haben. Wenn für die Richtlinie keine Absender-DNs angegeben sind, kann jeder Benutzer datengeschützte Nachrichten in die Warteschlange stellen, die das Zertifikat des Benutzers bereitstellt.

Absenderdefinierte Namen haben das folgende Format:

```
CN=Common Name,O=Organization,C=Country
```

Wichtig:

- Alle DNs müssen in Großbuchstaben angegeben werden. Alle Komponentennamenskennungen im DN müssen in der in der folgenden Tabelle angegebenen Reihenfolge angegeben werden:

Name der Komponente	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.

Name der Komponente	Wert
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.
- IBM WebSphere MQ AMS unterstützt nur Werte für definierte Namen aus dem Zeichensatz 'Lateinisches Alphabet 1'. Zum Erstellen von DNs mit Zeichen der Gruppe müssen Sie zuerst ein Zertifikat mit einem DN erstellen, der in UTF-8 -Codierung unter Verwendung von UNIX -Plattformen mit aktivierter UTF-8 -Codierung oder mit dem Dienstprogramm iKeyman erstellt wird. Anschließend müssen Sie eine Richtlinie von einer UNIX -Plattform mit aktivierter UTF-8 -Codierung erstellen oder das IBM WebSphere MQ AMS -Plug-in für WebSphere MQverwenden.

Zugehörige Konzepte

„Distinguished Names für Empfänger” auf Seite 327
Die definierten Namen (DN) des Empfängers geben Benutzer an, die berechtigt sind, Nachrichten aus einer Warteschlange abzurufen.

Distinguished Names für Empfänger

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 DNs müssen in Großbuchstaben angegeben werden. Alle Komponentennamenskennungen im DN müssen in der in der folgenden Tabelle angegebenen Reihenfolge angegeben werden:

Name der Komponente	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.
- Erweiterte Nachrichtensicherheit unterstützt nur Werte für definierte Namen aus dem Zeichensatz 'Lateinisches Alphabet 1'. Zum Erstellen von DNs mit Zeichen der Gruppe müssen Sie zuerst ein Zertifikat mit einem DN erstellen, der in UTF-8 -Codierung unter Verwendung von UNIX -Plattformen mit aktivierter UTF-8 -Codierung oder mit dem Dienstprogramm iKeyman erstellt wird. Anschließend müssen Sie eine Richtlinie von einer UNIX -Plattform mit aktivierter UTF-8 -Codierung erstellen oder das Erweiterte Nachrichtensicherheit -Plug-in für WebSphere MQverwenden.

Zugehörige Konzepte

„Distinguished Names für Absender” auf Seite 326
Die definierten Namen (DNs) des Absenders identifizieren Benutzer, die berechtigt sind, Nachrichten in eine Warteschlange einzureihen.

Sicherheitsrichtlinienattribute

Sie können Erweiterte Nachrichtensicherheit verwenden, um einen bestimmten Algorithmus oder eine bestimmte Methode zum Schutz der Daten auszuwählen.

Eine Sicherheitsrichtlinie ist ein konzeptionelles Objekt, das beschreibt, wie eine Nachricht verschlüsselt verschlüsselt und signiert wird. In der folgenden Tabelle finden Sie eine Übersicht über die Attribute von Sicherheitsrichtlinien in Erweiterte Nachrichtensicherheit:

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 Erweiterte Nachrichtensicherheit 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

Erweiterte Nachrichtensicherheit 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

Erweiterte Nachrichtensicherheit-Richtlinien für den Datenschutz beinhalten ein Datenschutzniveau (Quality of Protection, QOP).

Die drei Ebenen der Datenschutzqualität in Erweiterte Nachrichtensicherheit variieren je nach Verschlüsselungsalgorithmen, die zum Signieren und Verschlüsseln von Nachrichten 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.
- 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 keinen Signaturalgorithmus oder einen Verschlüsselungsalgorithmus festlegt, weist einen QOP von NONE auf. Erweiterte Nachrichtensicherheit bietet keinen Datenschutz für Warteschlangen, die über eine Richtlinie mit dem QOP NONE verfügen.

Sicherheitsrichtlinien verwalten

Eine Sicherheitsrichtlinie ist ein konzeptionelles Objekt, das beschreibt, wie eine Nachricht verschlüsselt verschlüsselt und signiert wird.

Alle in Zusammenhang mit Sicherheitsrichtlinien stehenden Verwaltungsaufgaben werden von der folgenden Speicherposition aus ausgeführt:

- Auf UNIX -Plattformen: `<MQInstallRoot>/bin`
- Auf Windows -Plattformen können Verwaltungstasks von jeder Position ausgeführt werden, da die Umgebungsvariable PATH bei der Installation aktualisiert wird.

Zugehörige Tasks

„Sicherheitsrichtlinien erstellen” auf Seite 330
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 ändern” auf Seite 330
Mit Erweiterte Nachrichtensicherheit können Sie Einzelheiten der Sicherheitsrichtlinien ändern, die Sie bereits definiert haben.

„Sicherheitsrichtlinien anzeigen und löschen” auf Seite 331

Mit dem Befehl `dspmqspl` können Sie eine Liste aller Sicherheitsrichtlinien oder Einzelheiten zu einer benannten Richtlinie in Abhängigkeit von den bereitgestellten Befehlszeilenparametern anzeigen.

„Sicherheitsrichtlinien entfernen” auf Seite 332
Wenn Sie Sicherheitsrichtlinien in Erweiterte Nachrichtensicherheit entfernen möchten, müssen Sie den Befehl `setmqspl` verwenden.

Sicherheitsrichtlinien 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 WebSphere MQ -Objekten entsprechen.
- Sie müssen über die erforderlichen `+connect +inq +chg` -Berechtigungen verfügen, um eine Sicherheitsrichtlinie zu erstellen. Die vollständige Syntax des Befehls zum Ändern der Berechtigung finden Sie in **`setmqaut`** .
- Stellen Sie sicher, dass Sie über die erforderlichen Berechtigungen zum Arbeiten mit WebSphere MQ -Warteschlangen und -Warteschlangenmanagern verfügen. Weitere Informationen finden Sie unter „OAM-Berechtigungen erteilen” auf Seite 334 .

Beispiel

Im Folgenden finden Sie ein Beispiel für die Erstellung einer Richtlinie für den Warteschlangenmanager `QMGR`. Die Richtlinie gibt an, dass Nachrichten mit dem SHA1-Algorithmus signiert und mit dem AES256-Algorithmus für Zertifikate mit dem definierten Namen (DN: CN=joe, O = IBM, C=US und DN: CN=jane, O = IBM, C = US) verschlüsselt werden sollen. Diese Richtlinie ist `MY.QUEUE` zugeordnet:

```
$ setmqspl -m QMGR -p MY.QUEUE -s SHA1 -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 mithilfe des DES-Algorithmus für Zertifikate mit folgenden DNs verschlüsselt werden sollen: CN=john, O = IBM, C=US und CN=jeff, O = IBM, C=US; und mit dem MD5-Algorithmus für das Zertifikat mit folgendem DN signiert werden sollen: CN=phil, O = IBM, C=US

```
$ setmqspl -m QMGR -p MY.OTHER.QUEUE -s MD5 -e DES -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 ändern

Mit Erweiterte Nachrichtensicherheit 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 erforderlichen `+connect +inq +chg` -Berechtigungen zum Erstellen von Sicherheitsrichtlinien verfügen. Die vollständige Syntax des Befehls zum Ändern der Berechtigung finden Sie in **setmqaut** .

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

Nachfolgend ein Beispiel für das Erstellen einer Richtlinie mit dem Namen MYQUEUE auf einem Warteschlangenmanager namens QMGR mit der Vorgabe, dass Nachrichten mit dem RC2-Algorithmus für Zertifikate mit folgendem DN verschlüsselt werden sollen:CN=bob,O=IBM,C=US; und mit dem SHA1-Algorithmus für Zertifikate mit folgendem DN signiert werden sollen:CN=jeff,O=IBM,C=US.

```
setmqspl -m QMGR -p MYQUEUE -e RC2 -s SHA1 -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 SHA1 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

Zugehörige Verweise

setmqspl

Sicherheitsrichtlinien 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 erforderlichen `+connect +inq +dsp` -Berechtigungen verfügen, die auf einen WS-Manager angewendet werden, um Sicherheitsrichtlinien anzuzeigen und zu erstellen. Die vollständige Syntax des Befehls zum Ändern der Berechtigung finden Sie in **setmqaut** .

Informationen zu diesem Vorgang

Die folgende Liste enthält die `dspmqspl` -Befehlsflags:

Tabelle 27. `dspmqspl` -Befehlsflags.

Befehlsmarkierung	Beschreibung
`-m`	Warteschlangenmanagername (**obligatorisch**).
`-p`	Richtlinienname.
`-export`	Durch das Hinzufügen dieses Flags wird eine Ausgabe generiert, die problemlos auf einen anderen WS-Manager angewendet werden kann.

Beispiel

In diesem Beispiel erstellen wir zwei Sicherheitsrichtlinien für `venus.queue.manager`:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s MD5 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s MD5 -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: MD5
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: MD5
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: MD5
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 MD5 -a "CN=signer1,O=IBM,C=US" -e NONE

dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

So importieren Sie eine Sicherheitsrichtlinie:

- Führen Sie auf Windows -Plattformen `policies.bat` aus.
- Auf UNIX -Plattformen:
  1. Melden Sie sich als Benutzer an, der zu der Verwaltungsgruppe von `mqm` WebSphere MQ gehört.
  2. Setzen Sie `. policies.sh` ab.

Zugehörige Verweise
Vollständige Liste der Attribute des Befehls 'dspmqspl'

Sicherheitsrichtlinien entfernen
Wenn Sie Sicherheitsrichtlinien in Erweiterte Nachrichtensicherheit 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 erforderlichen `+connect +inq +chg` -Berechtigungen zum Erstellen von Sicherheitsrichtlinien verfügen. Die vollständige Syntax des Befehls zum Ändern der Berechtigung finden Sie in **`setmqaut`** .

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

Systemwarteschlangenschutz

Systemwarteschlangen ermöglichen die Kommunikation zwischen WebSphere MQ und seinen Nebenanwendungen. Wenn ein Warteschlangenmanager erstellt wird, wird auch immer eine Systemwarteschlange erstellt, in der interne WebSphere MQ-Nachrichten und -Daten gespeichert werden. Sie können Systemwarteschlangen mit Erweiterte Nachrichtensicherheit schützen, so dass nur berechtigte Benutzer auf sie zugreifen oder sie entschlüsseln können.

Der Schutz der Systemwarteschlange folgt dem gleichen Muster wie der Schutz von regulären Warteschlangen. Siehe hierzu „Sicherheitsrichtlinien erstellen” auf Seite 330.

Wenn Sie den Systemwarteschlangenschutz auf Windows -Plattformen verwenden möchten, kopieren Sie die Datei `keystore.conf` in das folgende Verzeichnis:

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

Um Schutz für `SYSTEM.ADMIN.COMMAND.QUEUE`bereitzustellen, muss der Befehlsserver Zugriff auf die `keystore` und die `keystore.conf`haben, die Schlüssel und eine Konfiguration enthalten, damit der Befehlsserver auf Schlüssel und Zertifikate zugreifen kann. Alle Änderungen an der Sicherheitsrichtlinie von `SYSTEM.ADMIN.COMMAND.QUEUE` erfordern einen 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 Prüfung des definierten Namens (DN) des Unterzeichners nicht bestehen, nicht vom Befehlsserver ausgeführt und nicht an die Fehlerbehandlungswarteschlange Erweiterte Nachrichtensicherheit weitergeleitet. Nachrichten, die als Antworten an WebSphere MQ Explorer gesendet werden, werden nicht durch WebSphere MQ AMS geschützt.

Änderungen an den Erweiterte Nachrichtensicherheit -Sicherheitsrichtlinien erfordern einen Neustart des WebSphere MQ -Befehlsservers.

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
- 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
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM
- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- SYSTEM.CICS.INITIATION.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.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
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

OAM-Berechtigungen erteilen

Dateiberechtigungen berechtigen alle Benutzer, `setmqspl` -und `dspmqspl` -Befehle auszuführen. IBM Erweiterte Nachrichtensicherheit ist jedoch auf den Objektberechtigungsmanager (Object Authority Manager, OAM) angewiesen und jeder Versuch, diese Befehle von einem Benutzer auszuführen, der nicht zur Gruppe 'mqm' gehört, bei der es sich um die WebSphere MQ -Verwaltungsgruppe handelt oder der keine Berechtigung zum Lesen der erteilten Sicherheitsrichtlinieneinstellungen hat, 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
```

Befehls-und Konfigurationsereignisse

Mit Erweiterte Nachrichtensicherheit können Sie Befehle und Konfigurationsereignisnachrichten generieren, die protokolliert werden können und als Datensatz von Richtlinienänderungen für die Prüfung dienen.

Von WebSphere MQ generierte Befehls-und Konfigurationsereignisse sind Nachrichten im PCF-Format, die an dedizierte Warteschlangen gesendet werden.

Konfigurationsereignisnachrichten werden an das SYSTEM.ADMIN.CONFIG.EVENT -Warteschlange auf dem Warteschlangenmanager, auf dem das Ereignis auftritt

Befehlsereignisnachrichten werden an das SYSTEM.ADMIN.COMMAND.EVENT -Warteschlange auf dem Warteschlangenmanager, auf dem das Ereignis auftritt

Ereignisse werden unabhängig von den Tools generiert, die Sie zum Verwalten der Erweiterte Nachrichtensicherheit-Sicherheitsrichtlinien verwenden.

In Erweiterte Nachrichtensicherheit gibt es vier Ereignistypen, die von verschiedenen Aktionen in Sicherheitsrichtlinien generiert werden:

- „Sicherheitsrichtlinien erstellen” auf Seite 330, wobei zwei WebSphere MQ-Ereignisnachrichten generiert werden:
  - Ein Konfigurationsereignis
  - Ein Befehlsereignis
- „Sicherheitsrichtlinien ändern” auf Seite 330, wobei drei WebSphere MQ-Ereignisnachrichten generiert werden:
  - Ein Konfigurationsereignis, das alte Sicherheitsrichtlinienwerte enthält
  - Ein Konfigurationsereignis, das neue Sicherheitsrichtlinienwerte enthält.
  - Ein Befehlsereignis
- „Sicherheitsrichtlinien anzeigen und löschen” auf Seite 331, mit dem eine WebSphere MQ -Ereignisnachricht generiert wird:
  - Ein Befehlsereignis
- „Sicherheitsrichtlinien entfernen” auf Seite 332, wobei zwei WebSphere MQ-Ereignisnachrichten generiert werden:
  - Ein Konfigurationsereignis
  - Ein Befehlsereignis

Ereignisprotokollierung aktivieren und inaktivieren

Befehls- und Konfigurationsereignisse werden mit Hilfe der Warteschlangenmanagerattribute CONFIGEV und CMDEV gesteuert. 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 Konfigurationsereignisse inaktivieren möchten, setzen Sie CONFIGEV auf DISABLED . Sie können beispielsweise Konfigurationsereignisse mit dem folgenden MQSC-Befehl aktivieren:

```
ALTER QMGR CONFIGEV (ENABLED)
```

Befehlsereignisse

Um Befehlsereignisse zu aktivieren, setzen Sie CMDEV auf ENABLED . Um Befehlsereignisse für Befehle mit Ausnahme von DISPLAY MQSC-Befehlen und Inquire PCF-Befehlen zu aktivieren, setzen Sie den Parameter 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

Steuerung von Konfigurations-, Befehls- und Protokollierungsereignissen in WebSphere MQ

Befehlsereignisnachrichtenformat

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:

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.

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 WebSphere 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.

EventAccountingToken

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.

EventIdentityData

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.

EventApplName

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.

Command

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 WebSphere MQ 7.5 `cmqcfc.h` definiert.

Zurückgegeben: Immer.

CommandData enthält PCF-Elemente, die den PCF-Befehl enthalten.

Nachrichtenformat für Konfigurationsereignisse

Konfigurationsereignisse sind PCF-Nachrichten des Standardformats Erweiterte Nachrichtensicherheit .

Mögliche Werte für den MQMD-Nachrichtendeskriptor finden Sie unter 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 besteht aus der MQCFH-Struktur und der darauf folgenden Parameterstruktur. Mögliche MQCFH-Werte finden Sie unter 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 bei 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 WebSphere 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
Erweiterte Nachrichtensicherheit -Schutzrichtlinie. **1019** -ein numerischer Wert, der in WebSphere MQ 7.5 oder in der Datei `cmqc.h` definiert ist

Zurückgegeben: Immer.

PolicyName

Beschreibung: Der Name der Erweiterte Nachrichtensicherheit -Richtlinie

ID: **MQCA_POLICY_NAME** .

Datentyp: MQCFST.

Wert: **2112** -Ein numerischer Wert, der in WebSphere MQ 7.5 oder in der Datei `cmqc.h` definiert ist

Maximale Länge: MQ_OBJECT_NAME_LENGTH.

Zurückgegeben: Immer.

PolicyVersion

Beschreibung: Die Erweiterte Nachrichtensicherheit -Richtlinienversion.

ID: **MQIA_POLICY_VERSION**

Datentyp: MQCFIN

Wert **238** -ein numerischer Wert, der in WebSphere MQ 7.5 oder in der Datei `cmqc.h` definiert ist

Zurückgegeben: Immer

TolerateFlag

Beschreibung: Die Erweiterte Nachrichtensicherheit -Richtlinie für die Toleranzmarkierung.

ID: **MQIA_TOLERATE_UNPROTECTED**

Datentyp: MQCFIN

Wert **235** -ein numerischer Wert, der in WebSphere MQ 7.5 oder in der Datei `cmqc.h` definiert ist

Zurückgegeben: Immer.

SignatureAlgorithm

Beschreibung: Der Algorithmus der Erweiterte Nachrichtensicherheit -Richtliniensignatur.

ID: **MQIA_SIGNATURE_ALGORITHM**

Datentyp: MQCFIN

Wert: **236** -Ein numerischer Wert, der in WebSphere MQ 7.5 oder in der Datei `cmqc.h` definiert ist

Zurückgegeben: Sobald in der Erweiterte Nachrichtensicherheit-Richtlinie ein Signaturalgorithmus definiert ist

EncryptionAlgorithm

Beschreibung: Der Verschlüsselungsalgorithmus der Erweiterte Nachrichtensicherheit -Richtlinie.

ID: **MQIA_ENCRYPTION_ALGORITHM**

Datentyp: MQCFIN

Wert: **237** -Ein numerischer Wert, der in WebSphere MQ 7.5 oder in der Datei `cmqc.h` definiert ist

Zurückgegeben: Immer wenn ein Verschlüsselungsalgorithmus in der WebSphere MQ -Richtlinie definiert ist

SignerDNs

Beschreibung: Subjekt DistinguishedName der zulässigen Unterzeichner.

ID: **MQCA_SIGNER_DN**

Datentyp: MQCFSL

Wert: **2113** -Ein numerischer Wert, der in WebSphere MQ 7.5 oder in der Datei `cmqc.h` definiert ist

Maximale Länge: Längster Unterzeichner-DN in der Richtlinie, aber nicht mehr als MQ_DISTINGUISHED_NAME_LENGTH

Zurückgegeben: Bei jeder Definition in der WebSphere MQ -Richtlinie.

RecipientDNs

Beschreibung: Subjekt DistinguishedName der zulässigen Unterzeichner.

ID:	**MQCA_RECIPIENT_DN**
Datentyp:	MQCFSL
Wert:	**2114** -Ein numerischer Wert, der in WebSphere MQ 7.5 oder in der Datei `cmqc.h` definiert ist
Maximale Länge:	Längster Empfänger-DN in der Richtlinie, aber nicht mehr als MQ_DISTINGUISHED_NAME_LENGTH.
Zurückgegeben:	Bei jeder Definition in der WebSphere MQ -Richtlinie.

Probleme und Lösungen

In diesem Abschnitt wird beschrieben, wie Probleme gelöst werden, die bei einer Installation von IBMauftreten können. Verwenden Sie diese Informationen, um Probleme im Zusammenhang mit Erweiterte Nachrichtensicherheitzu identifizieren und zu beheben.

com.ibm.security.pkcsutil.PKCSException: Fehler beim Verschlüsseln des Inhalts

Fehler `com.ibm.security.pkcsutil.PKCSException: Error encrypting contents` weist darauf hin, dass IBM Erweiterte Nachrichtensicherheit Probleme beim Zugriff auf Verschlüsselungsalgorithmen hat.

Wenn der folgende Fehler von Erweiterte Nachrichtensicherheit zurückgegeben wird:

```
DRQJP0103E   The IBM WebSphere MQ Advanced Message Security Java interceptor failed to protect
message.
  com.ibm.security.pkcsutil.PKCSException: Error encrypting contents (java.security.Invalid
KeyException: Illegal key size or default parameters)
```

Prüfen Sie, ob die JCE-Sicherheitsrichtlinie in `JAVA_HOME/lib/security/local_policy.jar/*.policy` Zugriff auf die Signaturalgorithmen erteilt, die in der MQ-AMS-Richtlinie verwendet werden.

Wenn der zu verwendende Signaturalgorithmus nicht in Ihrer aktuellen Sicherheitsrichtlinie angegeben ist, laden Sie die richtige Java-Richtliniendatei von den folgenden Positionen herunter:

- IBM SDK-Richtliniendateien für Java 1.4.2.
- IBM SDK-Richtliniendateien für Java 5.0.
- IBM SDK-Richtliniendateien für Java 6.0.
- IBM SDK-Richtliniendateien für Java 7.0.

OSGi-Unterstützung

Zur Verwendung des OSGi-Bundles mit IBM Erweiterte Nachrichtensicherheit sind zusätzliche Parameter erforderlich.

Führen Sie den folgenden Parameter während des OSGi-Bundle-Starts aus:

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7
```

Wenn Sie verschlüsseltes Kennwort in der Datei "keystore.conf" verwenden, muss die folgende Anweisung hinzugefügt werden, wenn das OSGi-Bundle ausgeführt wird:

```
-Dorg.osgi.framework.system.packages.extra=com.ibm.security.pkcs7,com.ibm.misc
```

Einschränkung: IBM WebSphere MQ AMS unterstützt die Kommunikation nur mit MQ -Basis-Java-Klassen für Warteschlangen, die vor dem OSGi-Bundle geschützt sind.

Probleme beim Öffnen geschützter Warteschlangen bei der Verwendung von JMS

Es können verschiedene Probleme auftreten, wenn Sie geschützte Warteschlangen bei der Verwendung von IBM WebSphere MQ Advanced Message Security öffnen.

Sie führen JMS aus und empfangen den Fehler 2085 (MQRC_UNKNOWN_OBJECT_NAME) zusammen mit dem Fehler JMSMQ2008.

Sie haben überprüft, dass IBM WebSphere MQ Advanced Message Security wie unter „Schnelleinstieg für Java-Clients" auf Seite 300 beschrieben konfiguriert ist.

Eine mögliche Ursache ist, dass Sie eine andere Umgebung als IBM Java Runtime Environment verwenden. Dies ist eine bekannte Einschränkung, die in „Bekannte Einschränkungen" auf Seite 288beschrieben wird.

Sie haben die Umgebungsvariable AMQ_DISABLE_CLIENT_AMS nicht festgelegt.

Lösung des Problems

Es gibt vier Möglichkeiten, um dieses Problem zu lösen:

1. Starten Sie Ihre JMS -Anwendung unter einer unterstütztenIBMJava Runtime Environment (JRE).
2. Verschieben Sie Ihre Anwendung auf dieselbe Maschine, auf der Ihr Warteschlangenmanager ausgeführt wird, und lassen Sie sie über eine Verbindung im Bindungsmodus eine Verbindung herstellen.

   Eine Verbindung im Bindungsmodus verwendet plattformeigene native Bibliotheken für die Ausführung der IBM WebSphere MQ-API-Aufrufe. Daher wird der native AMS-Interceptor verwendet, um die AMS-Operationen auszuführen, und es gibt keine Abhängigkeit von den Funktionen der JRE.
3. Verwenden Sie einen MCA-Interceptor, da dies die Signierung und Verschlüsselung von Nachrichten ermöglicht, sobald sie im Warteschlangenmanager ankommen, ohne dass der Client eine AMS-Verarbeitung ausführen muss.

   Da der Schutz auf den Warteschlangenmanager angewendet wird, muss ein alternativer Mechanismus verwendet werden, um die Nachrichten, die vom Client zum Warteschlangenmanager übertragen werden, zu schützen. Meistens wird dies dadurch erreicht, dass die SSL/TLS-Verschlüsselung auf dem Serververbindungskanal konfiguriert wird, der von der Anwendung verwendet wird.
4. Legen Sie die Umgebungsvariable AMQ_DISABLE_CLIENT_AMS fest, wenn Sie nicht verwenden möchten.IBM WebSphere MQ Advanced Message Security

Weitere Informationen finden Sie im Abschnitt „Überwachung des Nachrichtenkanalagenten (MCA)" auf Seite 313.

Anmerkung: Es muss eine Sicherheitsrichtlinie für jede Warteschlange vorhanden sein, an die der MCA-Interceptor Nachrichten übergeben wird. Mit anderen Worten: Die Zielwarteschlange muss über eine AMS-Sicherheitsrichtlinie verfügen, die mit dem definierten Namen (DN) des Unterzeichners und des Empfängers übereinstimmt, der mit dem des Zertifikats übereinstimmt, das dem MCA-Interceptor zugeordnet ist. Dies ist der DN des Zertifikats, das durch die Eigenschaft `cms.certificate.channel.SYSTEM.DEF.SVRCONN` in dem vom Warteschlangenmanager verwendeten `keystore.conf` festgelegt ist.

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 zu geplanten Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zum Abrufen der Services von IBM WebSphere MQzu schreiben.

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 (http://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: