9.4

Protegendo o IBM MQ

Nota

Antes de usar estas informações e o produto que elas suportam, leia as informações em “Avisos” na página 705.

Esta edição se aplica à versão 9, liberação 4 do IBM® MQ e a todas as liberações e modificações subsequentes, até que seja indicado de outra forma em novas edições

Índice

Segurança do IBM MQ

A segurança é uma consideração importante tanto para desenvolvedores de aplicativos do IBM MQ quanto para administradores de sistema do IBM MQ. Como um mínimo absoluto, você deve assegurar que todo o hardware e software dentro da zona segura e nas estações de trabalho do operador estejam dentro de seu ciclo de vida de suporte, estejam atualizados com as atualizações de software obrigatórias e tenham as atualizações de segurança aplicadas prontamente.

Referências relacionadas

IBM Gerenciamento de vulnerabilidade de segurança

IBM Z e LinuxOne Portal de Segurança

Visão geral da segurança

Esta coleção de tópicos apresenta os conceitos de segurança do IBM MQ.

Os conceitos e mecanismos de segurança, conforme são aplicados a qualquer sistema de computador, são apresentados primeiro, seguido por uma discussão desses mecanismos de segurança, à medida que são implementados em IBM MQ.

Os aspectos comumente aceitos de segurança são os seguintes:

- “Identificação e autenticação” na página 7
- “Autorização” na página 9
- “Auditoria” na página 9
- “Sigilosidade” na página 10
- “Integridade de dados” na página 10

Mecanismos de Segurança são ferramentas técnicas e métodos que são utilizados para implementar serviços de segurança. Um mecanismo pode operar por conta própria, ou com outros, para fornecer um serviço específico. Exemplos de mecanismos de segurança comuns são os seguintes:

- “Criptografia” na página 11
- “Trechos de mensagens e assinaturas digitais” na página 13
- “Certificados Digitais” na página 13
- “Infraestrutura da Chave Pública (PKI)” na página 18

Ao planejar uma implementação do IBM MQ, considere quais mecanismos de segurança serão necessários para implementar os aspectos de segurança que são importantes para você. Para obter informações sobre o que considerar depois de ler esses tópicos, consulte “Planejando para seus requisitos de segurança” na página 90.

Identificação e autenticação

Identificação é a capacidade de identificar exclusivamente um usuário de um sistema ou um aplicativo que esteja sendo executado no sistema. *Autenticação* é a capacidade de provar que um usuário ou um aplicativo é realmente quem essa pessoa ou o que esse aplicativo diz ser.

Por exemplo, considere um usuário que entre no sistema com um ID de usuário e uma senha. O sistema usa o ID de usuário para identificar o usuário. O sistema autentica o usuário no momento do logon verificando se a senha fornecida está correta.

Identificação e autenticação em IBM MQ

Quando um aplicativo se conecta ao IBM MQ, uma identidade do usuário é sempre associada com a conexão A identidade do usuário é inicialmente o ID do usuário do Sistema Operacional que está

associado ao processo de aplicativo. Essa identidade geralmente é suficiente para aplicativos ligados localmente que são hospedados no mesmo sistema que o gerenciador de filas. No entanto, o gerenciador de filas também pode autenticar e modificar a identidade associada à conexão de várias maneiras. Autenticar a identidade associada a uma conexão é importante quando os aplicativos clientes que não podem necessariamente ser confiáveis se conectam a um gerenciador de filas por meio de uma rede.

A identidade associada a uma conexão de aplicativo com um gerenciador de filas do IBM MQ pode ser estabelecida usando qualquer um dos mecanismos a seguir:

- Quando um aplicativo se conecta a um gerenciador de filas, ele pode fornecer um ID do usuário e senha. O gerenciador de filas valida as credenciais com base em sua configuração Por exemplo, o ID do usuário e a senha podem ser passados para o Sistema Operacional do gerenciador de filas ou para o servidor LDAP, para serem autenticados.
- V 9.4.0 No IBM MQ 9.3.4, um aplicativo também pode fornecer um token de autenticação que ele obtém de um servidor de autenticação externo. Para obter mais informações sobre os tokens de autenticação, consulte “Trabalhando com tokens de autenticação.” na página 330
- Um canal do cliente pode ser configurado para usar a autenticação mútua TLS, se ele for configurado com um certificado digital válido. A autenticação de TLS pode ser combinada com uma regra de autenticação de canal (CHLAUTH) para associar um ID do usuário apropriado à conexão Para obter mais informações, consulte “Como o TLS fornece identificação, autenticação, confidencialidade e integridade” na página 20,
- Regras de autenticação de canal (CHLAUTH) podem substituir a identidade com base nas informações sobre a conexão. Por exemplo, uma regra de autenticação de canal pode configurar o ID do usuário associado a uma conexão baseada no endereço IP do cliente.
- O código de saída customizado pode configurar uma identidade com base em quaisquer critérios escolhidos.

Identidade e autenticação também são aplicáveis a canais entre dois gerenciadores de filas. Esses canais são conhecidos como canais de mensagens. Quando um canal de mensagens inicia, o agente do canal de mensagens (MCA) em cada extremidade do canal pode autenticar seu parceiro. Essa técnica é conhecida como *autenticação mútua*. Para o MCA emitente, essa é a garantia de que o parceiro ao qual ele está prestes a enviar mensagens é genuíno. Da mesma forma, o MCA de recebimento é assegurado que está prestes a receber mensagens de um parceiro genuíno.

Quando uma identidade tiver sido estabelecida e autenticada, se necessário, ela será usada pelo IBM MQ de várias maneiras:

- Importante, por padrão, quaisquer verificações subsequentes do “Autorização” na página 9 são feitas usando essa identidade. Por exemplo, se um aplicativo tentar colocar uma mensagem em uma fila, o gerenciador de filas confirmará que a identidade associada ao aplicativo possui autorização 'put' no objeto da fila.
- Além disso, cada mensagem pode conter informações de *contexto da mensagem* Essas informações são mantidas no descritor de mensagens (MQMD). O gerenciador de filas pode gerar automaticamente o contexto da mensagem quando um aplicativo coloca a mensagem em uma fila Como alternativa, o aplicativo poderá fornecer o contexto da mensagem se o ID do usuário associado ao aplicativo estiver autorizado a fazer isso. Essas informações de contexto em uma mensagem fornecem ao aplicativo que recebe as informações de mensagem sobre o originador da mensagem Ela contém, por exemplo, o nome do aplicativo que colocou a mensagem e o ID de usuário associados ao aplicativo.

Não repúdio

O objetivo geral do serviço de irrecusabilidade é poder provar que uma mensagem específica está associada a uma pessoa específica.

O serviço *não-repudiação* pode ser visto como uma extensão dos serviços de identificação e autenticação. Em geral, a não-repudiação é aplicada quando os dados são transmitidos eletronicamente; por exemplo, uma ordem ao corretor da bolsa para comprar ou vender ações, ou uma ordem a um banco para transferir fundos de uma conta a outra.

O serviço de não-repudiação pode conter mais de um componente, em que cada componente fornece uma função diferente. Caso o emitente da mensagem alguma vez se negue a enviá-la, o serviço de não-repudiação com a *prova de origem* pode fornecer ao receptor com completa certeza de que a mensagem foi enviada por aquele indivíduo específico. Caso o receptor da mensagem alguma vez se negue a enviá-la, o serviço de não-repudiação com a *prova de entrega* pode fornecer ao emitente com completa certeza de que a mensagem foi recebida por aquele indivíduo específico.

Na prática, provar com completa ou quase 100% de certeza, é uma tarefa difícil. No mundo real, nada é totalmente seguro. Gerenciamento de segurança está mais concentrado em gerenciar riscos a um nível que seja aceitável para os negócios. Em tal ambiente, uma expectativa mais realística do serviço de não-repudiação é ser capaz de fornecer provas que sejam admissíveis e que apóiem o seu caso em um tribunal de lei.

O não repúdio é um serviço de segurança relevante em um ambiente do IBM MQ porque o IBM MQ é um meio de transmitir dados eletronicamente. Por exemplo, você pode exigir provas contemporâneas de que uma mensagem específica foi enviada ou recebida por um aplicativo associado a um indivíduo em particular.

O IBM MQ com o Advanced Message Security não fornece um serviço de não repúdio como parte de sua função base. No entanto, esta documentação do produto traz sugestões sobre como é possível fornecer seu próprio serviço de não repúdio dentro de um serviço do IBM MQ elaborando seus próprios programas de saída.

Autorização

A *autorização* protege os recursos críticos em um sistema, limitando o acesso somente a usuários autorizados e seus aplicativos. Ele previne o uso não autorizado de um recurso ou o uso de um recurso de maneira não autorizada.

A autorização no IBM MQ

É possível usar a autorização para limitar o que indivíduos ou aplicativos específicos podem fazer em seu ambiente do IBM MQ.

Aqui estão alguns exemplos de autorização em um ambiente do IBM MQ:

- Permitir somente um administrador autorizado a emitir comandos para gerenciar os recursos do IBM MQ.
- Permitir que um aplicativo se conecte a um gerenciador de filas somente se o ID de usuário associado ao aplicativo estiver autorizado a fazê-lo.
- Permitir que um aplicativo abra somente as filas que são necessárias para sua função.
- Permitir que um aplicativo assine apenas os tópicos que forem necessários para sua função.
- Permitir que um aplicativo execute apenas operações em uma fila que sejam necessárias à sua função. Por exemplo, é possível que um aplicativo necessite somente procurar mensagens em uma fila especifica, e não colocar ou receber mensagens.

Para obter mais informações sobre como configurar autorização, consulte “Planejando a autorização” na página 94 e os subtópicos associados.

Auditoria

Auditoria é o processo de gravação e verificação de eventos para detectar se qualquer atividade inesperada ou desautorizada ocorreu, ou se nenhuma tentativa foi feita para executar essa atividade.

A auditoria no IBM MQ

O IBM MQ pode emitir mensagens de eventos para registrar que uma atividade incomum ocorreu.

Aqui estão alguns exemplos de auditoria em um ambiente do IBM MQ:

- Um aplicativo tenta abrir uma fila para que ele não está autorizado a abrir. Uma mensagem do evento de instrumentação é emitida. Inspecionando a mensagem do evento, você descobre que essa tentativa ocorreu e pode decidir qual ação é necessária.
- Um aplicativo tenta abrir um canal, mas a tentativa falha porque a conexão TLS não é permitida... Uma mensagem do evento de instrumentação é emitida. Inspecionando a mensagem do evento, você descobre que essa tentativa ocorreu e pode decidir qual ação é necessária.

Sigilosidade

O serviço de *confidencialidade* protege informações sensitivas de exposições não autorizadas.

Quando dados sensíveis são armazenados localmente, os mecanismos de controle de acesso podem ser suficientes para protegê-lo na hipótese de não ser possível ler os dados caso não possam ser acessados. Caso um nível mais alto de segurança seja necessário, os dados podem ser critografados.

Criptografe dados sensíveis quando são transmitidos em uma rede de comunicações, especialmente em uma rede tão insegura quanto a Internet. Em um ambiente de rede, os mecanismos de controle de acesso não são efetivos contra tentativas de intercepção de dados, como grampeamento de linha.

Confidencialidade em IBM MQ

É possível implementar a confidencialidade no IBM MQ, criptografando as mensagens.

A confidencialidade pode ser assegurada em um ambiente do IBM MQ, da seguinte forma:

- Depois que um MCA de envio recebe uma mensagem de uma fila de transmissão, o IBM MQ usa TLS para criptografar a mensagem antes de ela ser enviada pela rede ao MCA de recebimento. Na outra extremidade do canal, a mensagem é descodificada antes que o MCA receptor coloque-a em sua fila de destino.
- Enquanto as mensagens são armazenadas em uma fila local, os mecanismos de controle de acesso fornecidos pelo IBM MQ podem ser considerados suficientes para proteger os seus conteúdos contra divulgação não autorizada. No entanto, para um maior nível de segurança, é possível usar o Advanced Message Security para criptografar as mensagens armazenadas nas filas.
- z/OS As mensagens armazenadas nas filas locais podem ser criptografadas em repouso usando a criptografia do conjunto de dados do z/OS.

  Consulte a seção Confidencialidade para dados em repouso no IBM MQ for z/OS com criptografia do conjunto de dados. para obter informações adicionais.

Integridade de dados

O serviços *integridade dos dados* detecta se houve modificação não-autorizada dos dados.

Há duas maneiras nas quais os dados podem ser alterados: acidentalmente, por meio de erros de hardware ou transmissão ou em decorrência de um ataque deliberado. Muitos produtos de hardware e protocolos de transmissão possuem mecanismos para detectar e corrigir erros de hardware e de transmissão. O propósito de serviços de integridade de dados é detectar um ataque deliberado.

O serviço de integridade dos dados tem como objetivo somente detectar se algum dado foi modificado. Não tem a meta de restaurar dados ao seu estado original caso tenha sido modificado.

Mecanismos de controle de acesso podem contribuir para a integridade dos dados pois estes não podem ser modificados caso o acesso tenha sido negado. No entanto, como na confidencialidade, os mecanismos de controle de acesso não são efetivos em ambientes de rede.

Integridade de dados no IBM MQ

A integridade de dados pode ser assegurada em um ambiente do IBM MQ, da seguinte forma:

- É possível usar TLS para detectar se o conteúdo de uma mensagem foi deliberadamente modificado enquanto estava sendo transmitido por uma rede. No TLS, o algoritmo de trecho da mensagem fornece detecção de mensagens modificadas em trânsito.

  Todos os IBM MQ CipherSpecs fornecem um algoritmo de trecho da mensagem, exceto para TLS_RSA_WITH_NULL_NULL, que não fornece a integridade dos dados da mensagem.

  O IBM MQ detecta mensagens modificadas ao recebê-las, ao receber uma mensagem modificada, IBM MQ uma mensagem de erro AMQ9661 é gravada no registro de erros e o canal é interrompido.
- Enquanto as mensagens são armazenadas em uma fila local, os mecanismos de controle de acesso fornecidos pelo IBM MQ podem ser considerados suficientes para evitar a modificação intencional do conteúdo das mensagens.

  No entanto, para obter um nível de segurança maior, é possível usar o Advanced Message Security para detectar se o conteúdo de uma mensagem foi intencionalmente modificado entre o momento que a mensagem foi colocada na fila e o momento em que foi recuperada.

  Se uma mensagem modificada for detectada, o aplicativo tentando receber a mensagem receberá um código de retorno MQRC_SECURITY_ERROR (2063). Se o aplicativo estiver usando uma chamada MQGET , a mensagem também será movida para o SYSTEM.PROTECTION.ERROR.QUEUE fila.

Conceitos criptográficos

Esta coleção de tópicos descreve os conceitos de criptografia aplicáveis ao IBM MQ.

O termo *entidade* é usado para se referir a um gerenciador de fila, um IBM MQ MQI client, um usuário individual, ou qualquer outro sistema capaz de trocar mensagens.

Criptografia

Criptografia é o processo de conversão entre texto legível, chamado *texto simples*, e uma forma ilegível, chamada *texto cifrado*.

Isso ocorre conforme a seguir:

1. O emissor converte a mensagem de texto corrido para texto cifrado. Esta parte do processo é chamada de *criptografia* (às vezes *codificação*).
2. O texto cifrado é transmitido ao receptor.
3. O receptor converte a mensagem de texto cifrado de volta à sua forma de texto corrido. Esta parte do processo é chamada de *decriptografia* (às vezes *decifração*).

A conversação envolve uma seqüência de operações matemáticas que alteram a aparência da mensagem durante a transmissão, mas não afetam o conteúdo. As técnicas criptográficas podem assegurar confidencialidade e proteger mensagens contra visualização não autorizada (escuta), porque uma mensagem criptografada não é compreensível. Assinaturas digitais, que fornecem uma garantia da integridade da mensagem, usam técnicas de criptografia. Consulte a “Assinaturas digitais no SSL/TLS” na página 23 para obter mais informações.

Técnicas de criptografia envolvem um algoritmo geral, tornado específico pelo uso das chaves. Há duas classes de algoritmo:

- Aqueles que exigem que ambas as partes utilizem a mesma chave. Algoritmos que utilizam uma chave compartilhada são conhecidos como algoritmos *simétricos*. Figura 1 na página 12 ilustra a criptografia de chave simétrica.
- Aqueles que utilizam uma chave para criptografia e uma chave diferente para decriptografia. Um destes deve ser mantido secreto mas o outro pode ser público. Algoritmos que utilizam pares de chaves públicas e privadas são conhecidos como algoritmos *assimétricos*. Figura 2 na página 12 ilustra a criptografia de chave assimétrica, que também é conhecida como *criptografia de chave pública*.

Os algoritmos de criptografia e decriptografia utilizados podem ser públicos mas a chave secreta compartilhada e a chave privada devem ser mantidas secretas.

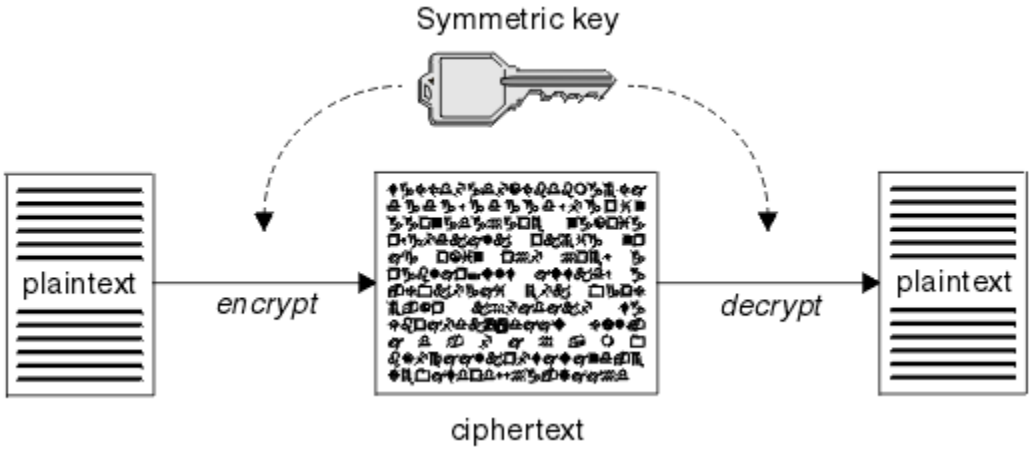


Figura 1. Criptografia de chave simétrica

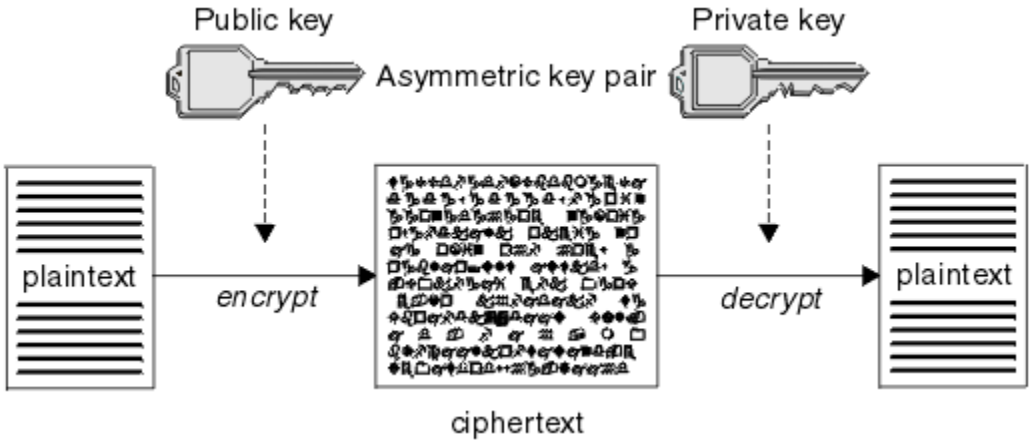


Figura 2. Criptografia de chave assimétrica

Figura 2 na página 12 mostra texto corrido criptografado com a chave pública do receptor e decriptografado com a chave privada do receptor. Somente o receptor pretendido tem a chave privada para decriptografar o texto cifrado. Observe que o emissor pode também criptografar mensagens com uma chave privada, que permite que qualquer um que tenha a chave pública do emissor decriptografe a mensagem, com a garantia de que a mensagem deve ter vindo do emissor.

Com algoritmos assimétricos, as mensagens são criptografadas com a chave pública ou privada mas podem ser decriptografados somente com a outra chave. Somente a chave privada é secreta, a chave pública pode ser conhecida por qualquer um. Com algoritmos simétricos, a chave compartilhada deve ser conhecida somente pelas duas partes. Isso chama-se *problema de distribuição de chave*. Algoritmos assimétricos são mais lentos mas têm a vantagem de que não há problema de distribuição de chave.

Outra terminologia associada com a criptografia é:

Ponto Forte
A força da criptografia é determinada pelo tamanho da chave. Algoritmos assimétricos exigem chaves grandes, por exemplo:

1024 bits	Chave assimétrica de força baixa
2048 bits	Chave assimétrica de força média
4096 bits	Chave assimétrica de força alta

As chaves simétricas são menores: as chaves de 256 bits fornecem criptografia avançada.

Algoritmo de cifra de bloco
Estes algoritmos criptografam dados por blocos. Por exemplo, o algoritmo RC2 do RSA Data Security Inc. usa blocos de 8 bytes de comprimento. Algoritmos de bloco são geralmente mais lentos do que algoritmos de fluxo.

Algoritmo de cifra de fluxo
Estes algoritmos operam em cada byte de dados. Algoritmos de fluxo são geralmente mais rápidos do que algoritmos de bloco.

Trechos de mensagens e assinaturas digitais

O trecho da mensagem é uma representação numérica de tamanho fixo do conteúdo de uma mensagem. O trecho da mensagem é calculado por uma função hash e pode ser criptografado, formando uma assinatura digital.

A função hash usada para calcular uma trecho da mensagem deve atender a dois critérios:

- Ela deve ser de uma maneira. Não deve ser possível reverter a função para encontrar a mensagem correspondente a um trecho de mensagem específico, a não ser testando todas as mensagens possíveis.
- Deve ser computacionalmente impossível encontrar duas mensagens que executem hash para a mesma compilação.

A compilação de mensagens é enviada junto com a própria mensagem. O destinatário pode gerar uma compilação para a mensagem e compará-la com a compilação do emissor. A integridade da mensagem é verificada quando os dois trechos da mensagem são os mesmos. Qualquer violação da mensagem durante a transmissão quase certamente resultará em uma compilação de mensagem diferente.

Um trecho da mensagem criado usando uma chave simétrica secreta é conhecido como um código de autenticação de mensagem (MAC), pois ele pode fornecer garantia de que a mensagem não foi modificada.

O emissor pode também gerar um trecho da mensagem e, em seguida, criptografar a compilação usando a chave privada de um par de chaves assimétricas, formando uma assinatura digital. A assinatura deve, então, ser decriptografada pelo receptor, antes de compará-la com uma compilação gerada localmente.

Conceitos relacionados
“Assinaturas digitais no SSL/TLS” na página 23
Uma assinatura digital é formada pela criptografia de uma representação de uma mensagem. A criptografia utiliza a chave privada do assinante e, para eficiência, geralmente opera em uma compilação de mensagens, ao invés de na mensagem em si.

Certificados Digitais

Os certificados digitais são protegidos contra personificação, certificando que uma chave pública pertence a uma entidade especificada. Eles são emitidos por uma autoridade de certificação.

Certificados digitais fornecem proteção contra identidades falsas, porque um certificado digital liga uma chave pública a seu proprietário, seja este um indivíduo, um gerenciador de filas ou alguma outra entidade. Certificados digitais também são conhecidos como certificados de chave pública, porque eles oferecem garantias sobre a propriedade de uma chave pública quando você utiliza um esquema de chave assimétrica. Um certificado digital contém a chave pública de uma entidade e é uma confirmação de que a chave pública pertence àquela entidade:

- Quando o certificado for para uma entidade individual, ele é chamado de *certificado pessoal* ou *certificado de usuário*.
- Quando o certificado for para uma Autoridade de Certificação, ele é chamado de *certificado de autoridade de certificação* ou *certificado de assinante*.

Se chaves públicas forem enviadas diretamente por seu proprietário a outra entidade, há um risco de que a mensagem possa ser interceptada e a chave pública ser substituída por outra. Isso é conhecido como *ataque humano intermediário* . A solução para este problema é trocar chaves públicas por meio

de terceiros confiáveis, o que proporcionará ao usuário uma garantia segura de que a chave pública realmente pertence à entidade com a qual você está se comunicando. Em vez de enviar sua chave pública diretamente, peça aos terceiros confiáveis para incorporá-la a um certificado digital. Os terceiros confiáveis que emitem certificados digitais são chamados de Autoridade de Certificação (CA), conforme descrito em "Autoridades de Certificação" na página 15.

O Que é um Certificado Digital

Os certificados digitais contêm partes específicas de informações, conforme determinado pelo padrão de X.509.

Certificados digitais usados pelo IBM MQ são compatíveis com o padrão X.509, que especifica as informações que são necessárias e o formato para enviá-las. X.509 é a estrutura de Autenticação que faz parte da série X.500 de padrões.

Os certificados digitais contêm, no mínimo, as seguintes informações sobre a entidade que está sendo certificada:

- A chave pública do proprietário
- O Nome Distinto do proprietário
- O Nome Distinto da CA que emitiu o certificado
- A data a partir da qual o certificado é válido
- A data de vencimento do certificado
- O número da versão do formato de dados do certificado conforme definido no X.509. A versão atual do padrão X.509 é Versão 3, e a maioria dos certificados está em conformidade com essa versão.
- Um número de série. Esse é um identificador exclusivo designado pelo CA que emitiu o certificado. O número de série é exclusivo dentro do CA que emitiu o certificado: não há dois certificados assinados pelo certificado de CA que têm o mesmo número de série.

Um certificado X.509 Versão 2 também contém um Identificador do Emissor e um Identificador de Assunto, e um certificado X.509 Versão 3 pode conter várias extensões. Algumas extensões de certificado, como a extensão de Restrição Básica, são *padrão*, mas outras são específicas à implementação. Uma extensão pode ser *crítica*, no caso em que um sistema deve ser capaz de reconhecer o campo. Se ele não reconhecer o campo, deverá rejeitar o certificado. Se uma extensão não for crítica, o sistema pode ignorá-la se não a reconhecer.

A assinatura digital em um certificado pessoal é gerada usando a chave privada do CA que assinou esse certificado. Qualquer pessoa que precisa verificar o certificado pessoal pode usar a chave pública do CA para fazer isso. O certificado do CA contém sua chave pública.

Os certificados digitais não contêm sua chave privada. Você deve manter sua chave privada em segredo.

Requisitos para certificados pessoais

O IBM MQ suporta certificados digitais compatíveis com o padrão X.509. Ele requer a opção de autenticação de cliente.

Como o IBM MQ é um sistema ponto a ponto, ele é visualizado como autenticação de cliente na terminologia SSL/TLS. Portanto, qualquer certificado pessoal usado para autenticação de SSL/TLS precisa permitir um uso principal de autenticação de cliente. Nem todos os certificados de servidor têm esta opção ativada, de forma que o fornecedor do certificado pode precisar ativar a autenticação de cliente no AC raiz do certificado seguro.

Além dos padrões que especificam o formato de dados para um certificado digital, há também os padrões para determinar se um certificado é válido. Essas normas foram atualizadas com o passar do tempo, a fim de evitar determinados tipos de violação de segurança. Por exemplo, os certificados mais antigos do X.509 versão 1 e 2 não indicam se o certificado pode ser legitimamente usado para assinar outros certificados. Era possível, portanto, para um usuário mal intencionado, obter um certificado pessoal de uma fonte legítima e criar novos certificados projetados para personificar outros usuários.

Ao usar certificados X.509 versão 3, o BasicConstraints e as extensões de certificado KeyUsage são usados para especificar quais certificados podem assinar legitimamente outros certificados. O padrão

IETF RFC 5280 especifica uma série de regras de validação de certificado que o software de aplicativo compatível deve implementar para impedir ataques de personificação. Um conjunto de regras de certificado é conhecido como uma política de validação de certificado.

Para obter mais informações sobre as políticas de validação de certificado no IBM MQ, consulte "Políticas de validação de certificado no IBM MQ" na página 46.

Autoridades de Certificação

Uma autoridade de certificação (CA) é um terceiro confiável que emite certificados digitais para fornecer a garantia de que a chave pública de uma entidade verdadeiramente pertença àquela entidade.

As funções de uma CA são:

- Ao receber um pedido de um certificado digital, verifique a identidade do solicitante antes de construir, assinar e devolver o certificado pessoal
- Fornecer a chave pública da própria CA em seu certificado de CA
- Publicar listas de certificados que não são mais confiáveis em uma CRL (Lista de Revogação de Certificados). Para obter mais informações, consulte "Trabalhando com Certificados Revogados" na página 345
- Para fornecer acesso ao status de revogação de certificado, operando um respondente servidor de respondente do OCSP

Nomes Distintos

O DN (Distinguished Name) identifica de modo exclusivo uma entidade em um certificado X.509.

Atenção: Apenas os atributos na tabela a seguir podem ser usados em um filtro SSLPEER. Os DNs do certificado podem conter outros atributos, mas a filtragem não é permitida nesses atributos.

Tabela 1. Tipos de atributos localizados no DN que podem ser usados em um filtro SSLPEER

Tipo de atributo	**Descrição**
SERIALNUMBER	Número de série do certificado
MAIL	Endereço de e-mail
Deprecated E	Endereço de e-mail (descontinuado na preferência para MAIL)
UID ou USERID	Identificador de usuários
CN	Nome Comum
T	Título
OU	Nome de Unidade Organizacional
DC	Componente de domínio
O	Nome da organização
STREET	Rua / Primeira linha do endereço
L	Nome da localidade
ST (ou SP ou S)	Nome do estado ou região
PC	Código Postal / Código de Endereçamento Postal
C	País
UNSTRUCTUREDNAME	Nome do host
UNSTRUCTUREDADDRESS	endereço IP
DNQ	Qualificador de Nome Distinto

O padrão X.509 define outros atributos que normalmente não fazem parte do DN, mas podem fornecer extensões opcionais para o certificado digital.

O padrão X.509 faz com que um DN seja especificado em formato de cadeia. Por exemplo:

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

O Nome Comum (CN) pode descrever um usuário individual ou qualquer outra entidade, por exemplo, um servidor da Web.

O DN pode conter diversos atributos OU e DC. Apenas uma instância de cada um dos outros atributos é permitida. A ordem das entradas de OU é significante: ela especifica uma hierarquia de nomes de Unidades Organizacionais, com a unidade de mais alto nível primeiro. A ordem das entradas DC também é significativa.

IBM MQ tolera certos DNs malformados. Para obter mais informações, consulte as regras do IBM MQ para valores SSLPEER.

Conceitos relacionados

"O Que é um Certificado Digital" na página 14
Os certificados digitais contêm partes específicas de informações, conforme determinado pelo padrão de X.509.

Obtendo certificados pessoais a partir de uma autoridade de certificação

É possível obter um certificado a partir de uma autoridade de certificação (CA) externa confiável.

Você obtém um certificado digital enviando informações a um CA na forma de uma solicitação de certificado. O padrão X.509 define um formato para estas informações, mas alguns CAs têm seu próprio formato. Solicitações de certificado são geralmente geradas pela ferramenta de gerenciamento de certificados que seu sistema usa; por exemplo:

- ALW Os comandos **runmqakm** e V 9.4.0 V 9.4.0 **runmqktool** no AIX, Linux, and Windows..
- z/OS RACF no z/OS.

As informações contêm seu Nome Distinto e sua chave pública. Quando sua ferramenta de gerenciamento de certificados gera seu pedido de certificado, também gera sua chave privada, que você deve manter segura. Nunca distribua sua chave privada.

Quando a CA recebe seu pedido, a autoridade verifica sua identidade antes de construir o certificado e devolvê-lo a você como um certificado pessoal.

Figura 3 na página 16 ilustra o processo de obtenção de um certificado digital de uma CA.

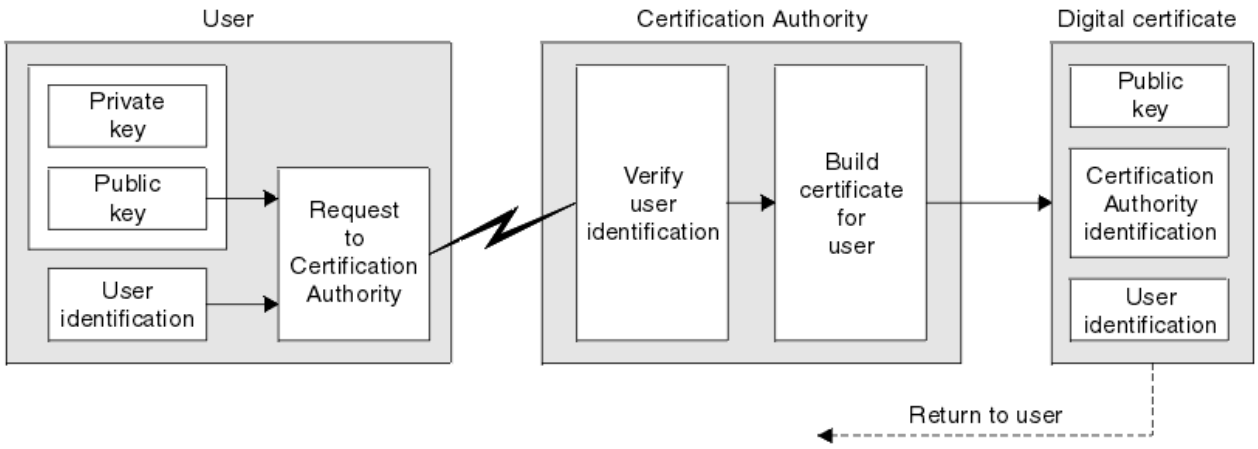


Figura 3. Obtendo um certificado digital

No diagrama:

- A identificação de usuário inclui o seu Nome distinto do assunto.
- A identificação de autoridade de certificação inclui o Nome distinto da autoridade de certificação que está emitindo o certificado.

Os certificados digitais contêm campos adicionais diferentes dos mostrados no diagrama. Para obter mais informações sobre os outros campos em um certificado digital, consulte "O Que é um Certificado Digital" na página 14.

Como Funcionam as Cadeias de Certificados

Quando você receber o certificado de outra entidade, você pode precisar utilizar uma *cadeia de certificados* para obter o certificado *CA raiz*.

A cadeia de certificados, também conhecida como *caminho de certificação*, é uma lista de certificados utilizada para autenticar uma entidade. A cadeia, ou caminho, começa com o certificado daquela entidade, e cada certificado na cadeia é assinado pela entidade identificada pelo próximo certificado na cadeia. A cadeia termina com um certificado de CA raiz. O certificado de autoridade de certificação raiz é sempre assinado pela própria autoridade de certificação (CA). As assinaturas de todos os certificados na cadeia devem ser verificadas até que o certificado de CA raiz seja alcançado.

Figura 4 na página 17 ilustra um caminho de certificação do proprietário do certificado para a CA raiz, onde a cadeia de confiança começa.

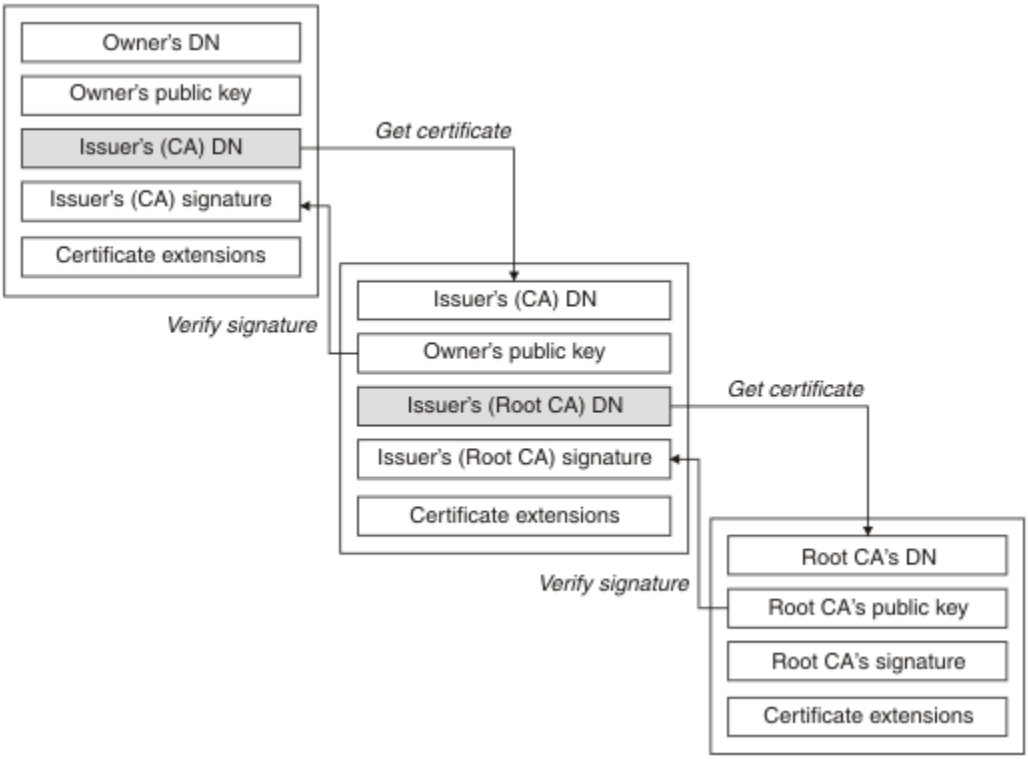


Figura 4. Cadeia de confiança

Cada certificado pode conter uma ou mais extensões. Um certificado pertencente a uma CA geralmente contém uma extensão BasicConstraints com o sinalizador isCA configurado para indicar que é permitido que ele assine outros certificados.

Quando os Certificados Não São Mais Válidos

Os certificados digitais podem vencer ou serem revogados.

Certificados digitais também são emitidos por um período fixo e não são válidos depois de suas datas de expiração.

Certificados podem ser revogados por várias razões, incluindo:

- O proprietário mudou para uma organização diferente.
- A chave privada não é mais secreta.

O IBM MQ pode verificar se um certificado foi revogado enviando uma solicitação para um respondente do Online Certificate Status Protocol (OCSP) (no AIX, Linux, and Windows somente). Como alternativa, eles podem acessar uma lista de revogação de certificado (CRL) em um servidor LDAP. A revogação do OCSP e as informações da CRL são publicadas por uma autoridade de certificação. Para obter mais informações, consulte "Trabalhando com Certificados Revogados" na página 345.

Infraestrutura da Chave Pública (PKI)

O PKI (Public Key Infrastructure) é um sistema de recursos, políticas, e serviços que suportam a utilização de criptografia de tecla pública para autenticar as partes envolvidas na transação.

Não há nem um único padrão que define os componentes de uma infraestrutura de chave pública, mas uma infraestrutura de chave pública geralmente inclui autoridades de certificação (CAs) e Autoridades de registro (RAs). Os CAs fornecem os serviços a seguir:

- Emissão de certificados digitais
- Validação de certificados digitais
- Revogação de certificados digitais
- Distribuição de teclas públicas

Os padrões X.509 fornecem a base para a infraestrutura de chave pública padrão de mercado.

Consulte "Certificados Digitais" na página 13 para obter mais informações sobre certificados digitais e autoridades de certificação (CAs). Os RAs verificam a informações fornecidas quando os certificados digitais são exigidos. Se o RA verificar a informação, o CA pode emitir um certificados digital ao solicitante.

Um PKI também pode fornecer as ferramentas para o gerenciamento de certificados digitais ou teclas públicas. Um PKI, às vezes, é descrito como uma *hierarquia de confiança* para o gerenciamento de certificados digitais, mas a maioria das definições incluem serviços adicionais. Algumas definições incluem serviços de criptografia e de assinatura digital, mas não são essenciais para a operação de um PKI.

Protocolos de segurança criptográficos: TLS

Os protocolos de criptografia fornecem conexões seguras, permitindo que dois grupos de comuniquem com privacidade e integridade de dados. O protocolo Transport Layer Security (TLS) surgiu desse do Secure Sockets Layer (SSL). O IBM MQ suporta o TLS.

Os objetivos principais dos dois protocolos é fornecer confidencialidade, (às vezes referida como *privacidade*), integridade de dados, identificação e autenticação, usando certificados digitais.

Embora os dois protocolos sejam semelhantes, as diferenças são suficientemente significantes que o SSL 3.0 e as várias outras versões do TLS não interoperam.

Conceitos relacionados

"Protocolos de segurança TLS no IBM MQ" na página 25
O IBM MQ suporta o protocolo da Segurança da Camada de Transporte (TLS) para fornecer segurança em nível de link para canais de mensagens e canais do MQI.

Conceitos de TLS (Transport Layer Security)

O protocolo TLS permite que duas partes identifiquem e autentiquem uma a outra e se comuniquem com confidencialidade e integridade de dados. O protocolo TLS surgiu do protocolo Netscape SSL 3.0, mas o TLS e o SSL não interoperam.

O protocolo TLS fornece segurança de comunicações sobre a Internet e permite que os aplicativos cliente/servidor se comuniquem de uma forma que seja confidencial e confiável. Os protocolo possuem duas camadas: um Protocolo de Registro e um Protocolo de Handshake e, eles são dispostos em camadas sobre um protocolo de transporte como TCP/IP. Ambos usam técnicas de criptografia assimétrica e simétrica.

Uma conexão do TLS é iniciada por um aplicativo, que se torna o cliente do TLS. O aplicativo que recebe a conexão se torna o servidor do TLS. Cada nova sessão se inicia com um handshake, conforme definido pelos protocolos do TLS.

Uma lista integral de CipherSpecs suportados por IBM MQ é fornecida em "Ativando CipherSpecs" na página 427.

Para obter mais informações sobre o protocolo SSL, consulte as informações fornecidas em https://developer.mozilla.org/docs/Mozilla/Projects/NSS. Para obter mais informações sobre o protocolo TLS, consulte as informações fornecidas pelo Grupo de Trabalho do TLS no website do Internet Engineering Task Force em https://www.ietf.org.

Uma visão geral do handshake SSL/TLS

O handshake SSL/TLS permite que o cliente e o servidor TLS estabeleçam as chaves secretas com as quais se comunicam.

Esta seção fornece um resumo das etapas que permitem que o cliente e o servidor TLS se comuniquem entre si.

- Aceitar a versão do protocolo a ser usada.
- Selecionar algoritmos criptográficos, que são descritos em .
- Autenticar um ao outro com a troca e validação de certificados digitais.
- Utilizar técnicas de criptografia assimétrica para gerar uma chave compartilhada secreta, que evita o problema de distribuição de chaves. O TLS em seguida, usa a chave compartilhada para a criptografia simétrica das mensagens, que é mais rápida que a criptografia assimétrica.

Para obter mais informações sobre a importação de certificados, consulte a seção relevante para sua plataforma em .

Na visão geral, as etapas envolvidas no handshake do TLS são como a seguir:

1. O cliente do TLS envia uma mensagem "Olá, cliente" que lista as informações de criptografia como a versão do TLS e, na ordem de preferência do cliente, os CipherSuites suportados pelo cliente. A mensagem também contém uma cadeia de bytes aleatória que é utilizada em cálculos subsequentes. O protocolo permite que o "client hello" inclua métodos de compactação de dados suportados pelo cliente.
2. O servidor do TLS responde com uma mensagem "Olá, servidor" que contém o CipherSuite escolhido pelo servidor na lista fornecida pelo cliente, o ID da sessão e outra sequência de bytes aleatória. O servidor também envia seu certificado digital. Se o servidor exigir um certificado digital para a autenticação do cliente, o servidor envia um "pedido de certificado de cliente" que inclui uma lista dos tipos de certificados suportados e os Nomes Distintos de Autoridades de Certificação (CAs) aceitáveis.
3. O cliente do TLS verifica o certificado digital do servidor. Para obter informações adicionais, consulte "Como o TLS fornece identificação, autenticação, confidencialidade e integridade" na página 20.
4. O cliente do TLS envia a sequência de bytes aleatória que permite que ambos o cliente e o servidor calculem a chave secreta a ser usada para criptografar dados de mensagem subsequentes. A própria cadeia de bytes aleatória é criptografada com a chave pública do servidor.
5. Se o servidor do TLS enviou uma "solicitação de certificado de cliente", o cliente enviará uma sequência de bytes aleatória criptografada com a chave privada do cliente, junto com o certificado digital do cliente ou um "alerta de certificado não digital". Este alerta é apenas um aviso, mas com algumas implementações, o protocolo de reconhecimento falha em caso de obrigatoriedade da autenticação do cliente.

6. O servidor do TLS verifica o certificado de cliente. Para obter informações adicionais, consulte “Como o TLS fornece identificação, autenticação, confidencialidade e integridade” na página 20.
7. O cliente do TLS envia ao servidor uma mensagem de "concluído", que é criptografada com a chave secreta, indicando que a parte do handshake do cliente está concluída.
8. O servidor do TLS envia ao cliente uma mensagem de "concluído", que é criptografada com a chave secreta, indicando que a parte do servidor do handshake está concluída.
9. Para a duração da sessão do TLS, o servidor e o cliente agora podem trocar mensagens que são criptografadas simetricamente com a chave secreta compartilhada.

Figura 5 na página 20 ilustra o handshake do TLS.

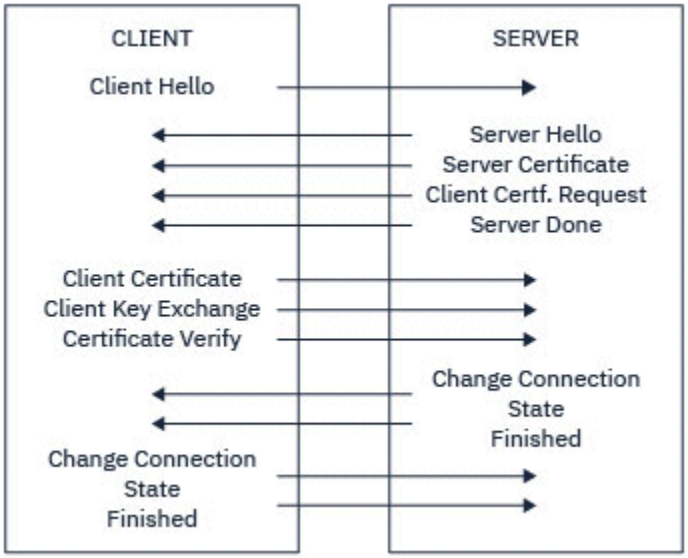


Figura 5. Visão geral do handshake do TLS

Como o TLS fornece identificação, autenticação, confidencialidade e integridade

Durante a autenticação do cliente e do servidor, existe uma etapa que requer que os dados sejam criptografados com uma das chaves de um par de chaves assimétricas e decriptografados com a outra chave do par. Um trecho da mensagem é usado para fornecer integridade.

Para obter uma visão geral das etapas envolvidas no handshake TLS, consulte “Uma visão geral do handshake SSL/TLS” na página 19..

Como o TLS fornece autenticação

Para autenticação do servidor, o cliente utiliza a chave pública do servidor para criptografar os dados que são utilizados para calcular a chave secreta. O servidor poderá gerar a chave secreta somente se puder descriptografar esses dados com a chave privada correta. A própria sequência de bytes aleatória é criptografada com a chave pública do servidor (etapa “4” na página 19 na visão geral).

Para a autenticação do cliente, o servidor utiliza a chave pública do certificado do cliente para decriptografar os dados enviados pelo cliente durante a etapa “5” na página 19 do protocolo de reconhecimento. A troca de mensagens concluídas que são criptografadas com a chave secreta (etapas “7” na página 20 e “8” na página 20 na visão geral) confirma que a autenticação está completa.

Se alguma das etapas de autenticação falhar, o protocolo de reconhecimento falhará e a sessão será encerrada.

A troca de certificados digitais durante o handshake TLS faz parte do processo de autenticação. Para obter informações adicionais sobre como os certificados fornecem proteção contra personificação,

consulte as informações relacionadas. Os certificados requeridos são os seguintes, em que CA X emite o certificado para o cliente TLS e CA Y emite o certificado para o servidor TLS:

Somente para autenticação de servidor, o servidor TLS precisa:

- O certificado pessoal emitido para o servidor pela autoridade de certificação Y
- Da chave privada do servidor

e o cliente TLS precisa:

- O certificado de CA para CA Y

Se o servidor TLS requerer autenticação de cliente, o servidor verificará a identidade do cliente verificando o certificado digital do cliente com a chave pública para a CA que emitiu o certificado pessoal para o cliente, neste caso, a CA X. Para autenticação de servidor e cliente, o servidor precisa:

- O certificado pessoal emitido para o servidor pela autoridade de certificação Y
- Da chave privada do servidor
- O certificado de CA para CA X

e o cliente precisa:

- O certificado pessoal emitido para o cliente pela autoridade de certificação X
- Da chave privada do cliente
- O certificado de CA para CA Y

O cliente e o servidor TLS podem precisar de outros certificados de CA para formarem uma cadeia de certificados para o certificado de CA raiz. Para obter informações adicionais sobre as cadeias de certificados, consulte as informações relacionadas.

O que Acontece Durante a Verificação de Certificado

Conforme observado nas etapas "3" na página 19 e "6" na página 20 da visão geral, o cliente TLS verifica o certificado do servidor e o servidor TLS verifica o certificado do cliente. Existem quatro aspectos para esta verificação:

1. A assinatura digital é verificada (consulte "Assinaturas digitais no SSL/TLS" na página 23).
2. A cadeia de certificados é verificada; é necessário ter certificados de autoridade de certificação intermediária (consulte "Como Funcionam as Cadeias de Certificados" na página 17).
3. As datas de expiração e de ativação e o período de validade são verificados.
4. O status de revogação do certificado é verificado (consulte "Trabalhando com Certificados Revogados" na página 345).

Reconfiguração de Chave Secreta

Durante um handshake TLS, uma *chave secreta* é gerada para criptografar dados entre o cliente e o servidor TLS. A chave secreta é utilizada em uma fórmula matemática que é aplicada aos dados para transformar o texto puro em texto criptografado ilegível, e texto criptografado em texto puro.

A chave secreta é gerada a partir do texto aleatório enviado como parte da handshake, e é usada para criptografar um texto simples para texto cifrado. A chave secreta também é utilizada no algoritmo MAC (Message Authentication Code), que é utilizado para determinar se uma mensagem foi alterada. Consulte a "Trechos de mensagens e assinaturas digitais" na página 13 para obter mais informações.

Caso a chave secreta seja descoberta, o texto puro de uma mensagem poderia ser decifrado a partir do texto criptografado, ou o resumo da mensagem poderia ser calculado, permitindo que mensagens sejam alteradas sem que isso seja detectado. Mesmo para um algoritmo complexo, o texto puro pode eventualmente ser descoberto aplicando todas as transformações matematicamente possíveis ao texto criptografado. Para minimizar a quantidade de dados que podem ser decifrados ou alterados caso a chave secreta seja descoberta, a chave secreta pode ser renegociada periodicamente. Quando a chave secreta

for renegociada, a chave secreta anterior não poderá mais ser usada para decriptografar dados que foram criptografados com a nova chave secreta.

Como o TLS fornece confidencialidade

O TLS usa uma combinação de criptografia simétrica e assimétrica para assegurar a privacidade da mensagem. Durante o handshake TLS, o cliente e o servidor TLS concordam em relação a um algoritmo de criptografia e uma chave secreta compartilhada que serão usados somente para uma sessão. Todas as mensagens transmitidas entre o cliente e o servidor TLS serão criptografadas usando esse algoritmo e essa chave, o que assegura que a mensagem continuará sendo privada se for interceptada. Como o TLS usa criptografia assimétrica ao transportar a chave secreta compartilhada, não há nenhum problema de distribuição de chaves. Para obter mais informações sobre técnicas de criptografia, consulte “Criptografia” na página 11.

Como o TLS fornece integridade

O TLS fornece integridade de dados calculando um trecho da mensagem. Para obter informações adicionais, consulte “Integridade de dados de mensagens” na página 484.

O uso de TLS assegura integridade de dados, contanto que o CipherSpec na definição de canal use um algoritmo hash, conforme descrito na tabela em “Ativando CipherSpecs” na página 427.

Especificamente, se a integridade de dados for um problema, você deve evitar a escolha de um CipherSpec cujo algoritmo hash esteja listado como "Nenhum". O uso de MD5 também é fortemente desencorajado, pois isso agora é muito antigo e não é mais seguro para a maioria dos propósitos práticos.

CipherSpecs e CipherSuites

Os protocolos de segurança criptográficos devem concordar sobre os algoritmos usados por uma conexão segura. CipherSpecs e CipherSuites definem combinações específicas de algoritmos.

Um CipherSpec identifica uma combinação de algoritmo de criptografia e algoritmo de código de autenticação de mensagem (MAC). As extremidades de uma conexão TLS devem concordar com o mesmo CipherSpec para serem capazes de se comunicar.

O IBM MQ suporta protocolos TLS1.3 e TLS1.2 e CipherSpecs. No entanto, será possível ativar CipherSpecs descontinuadas, se você precisar fazer isso.

Consulte “Ativando CipherSpecs” na página 427 para obter informações sobre:

- CipherSpecs suportados pelo IBM MQ
- Como ativar especificações de código descontinuadas do SSL 3.0 e do TLS 1.0

Importante: Ao lidar com canais do IBM MQ, você usa um CipherSpec. Ao lidar com canais do Java, canais do JMS ou canais do MQTT, especifique um CipherSuite.

Para obter mais informações sobre CipherSpecs, consulte “Ativando CipherSpecs” na página 427.

Um CipherSuite é um conjunto de algoritmos criptográficos usados por uma conexão TLS. Um conjunto compreende três algoritmos distintos:

- O algoritmo de troca de chave e autenticação, usado durante o handshake
- O algoritmo de criptografia, utilizado para codificar os dados
- O algoritmo MAC (Message Authentication Code), utilizado para gerar a compilação de mensagem

Há várias opções para cada componente do conjunto, mas somente certas combinações são válidas quando especificadas para uma conexão TLS. O nome de um CipherSuite válido define a combinação de algoritmos utilizada. Por exemplo, o CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA especifica:

- O algoritmo de troca de chaves e autenticação RSA
- O algoritmo de criptografia AES, usando uma chave de 128 bits e o modo de encadeamento de blocos cifrados (CBC)

- O Código de Autenticação de Mensagem SHA-1 (MAC)

Assinaturas digitais no SSL/TLS

Uma assinatura digital é formada pela criptografia de uma representação de uma mensagem. A criptografia utiliza a chave privada do assinante e, para eficiência, geralmente opera em uma compilação de mensagens, ao invés de na mensagem em si.

Assinaturas digitais variam com os dados sendo assinados, diferente de com assinaturas manuscritas, que não dependem do conteúdo do documento sendo assinado. Se duas mensagens diferentes forem assinadas digitalmente pela mesma entidade, as duas assinaturas diferirão, mas ambas poderão ser verificadas com a mesma chave pública, ou seja, a chave pública da entidade que assinou as mensagens.

As etapas do processo de assinatura digital são as seguintes:

1. O emissor calcula uma compilação de mensagens, e então a criptografa, utilizando a chave privada do emissor, formando a assinatura digital.
2. O emissor transmite a assinatura digital com a mensagem.
3. O receptor decriptografa a assinatura digital utilizando a chave pública do emissor, gerando novamente a compilação de mensagens do emissor.
4. O receptor calcula uma compilação de mensagem recebida de dados de mensagem e verifica se as duas compilações são as mesmas.

Figura 6 na página 23 ilustra este processo.

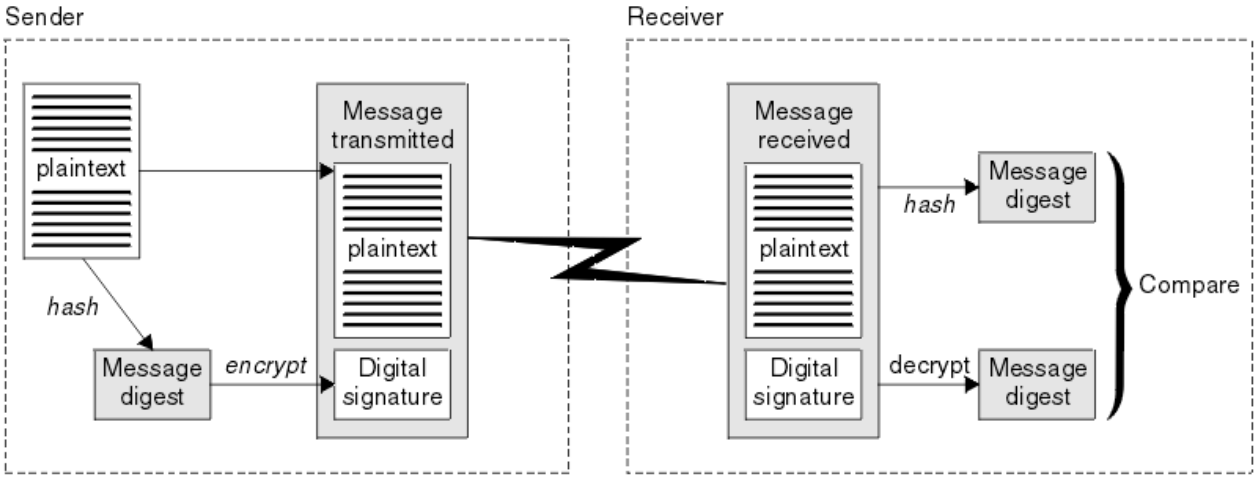


Figura 6. O processo de assinatura digital

Se a assinatura digital for verificada, o receptor saberá que:

- A mensagem não foi modificada durante a transmissão.
- A mensagem foi enviada pela entidade que declara tê-la enviado.

Assinaturas digitais são parte dos serviços de integridade e autenticação. Assinaturas digitais também proporcionam prova de origem. Somente o emissor conhece a chave privada, que fornece forte evidência de que o emissor é o originador da mensagem.

Nota: Você pode também criptografar a própria mensagem, que protegerá a confidencialidade das informações da mensagem.

Federal Information Processing Standards

O governo dos EUA produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. O National Institute for Standards and Technology (NIST) é um órgão importante ligado a sistemas e segurança de TI. O NIST produz recomendações e normas, inclusive Federal Information Processing Standards (FIPS).

Um dessas normas significativa é FIPS 140-2, que requer o uso de algoritmos criptográficos fortes. FIPS 140-2 também especifica requisitos para algoritmos hashing a serem usados para proteger pacotes contra modificação em trânsito.

Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o IBM Crypto for C (ICC) certificado e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na NIST CMVP modules in process list.

O IBM MQ Operator 3.2.0 e a imagem do contêiner do gerenciador de filas 9.4.0.0 em diante são baseados no UBI 9 A conformidade do FIPS 140-3 está pendente atualmente e seu status pode ser visualizado procurando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" no NIST CMVP modules in process list.

O IBM MQ fornece suporte a FIPS 140-2 quando tiver sido configurado para tal.

Ao longo do tempo, analistas desenvolvem ataques contra algoritmos de criptografia e hashing existentes. Novos algoritmos são adotados para resistir a esses ataques. FIPS 140-2 é atualizada periodicamente para considerar essas mudanças.

Conceitos relacionados

“Criptografia do Conjunto B da Agência Nacional de Segurança (NSA)” na página 24
O governo dos Estados Unidos da América produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. A Agência Nacional de Segurança dos Estados Unidos (NSA) recomenda um conjunto de algoritmos criptográficos interoperável em seu padrão do Conjunto B.

Criptografia do Conjunto B da Agência Nacional de Segurança (NSA)

O governo dos Estados Unidos da América produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. A Agência Nacional de Segurança dos Estados Unidos (NSA) recomenda um conjunto de algoritmos criptográficos interoperável em seu padrão do Conjunto B.

O padrão do Conjunto B especifica um modo de operação no qual somente um conjunto específico de algoritmo criptográfico seguros são usados. O padrão do Conjunto B especifica:

- O algoritmo de criptografia (AES)
- O algoritmo de troca de chave (Diffie-Hellman da curva elíptica, também conhecido como ECDH)
- O algoritmo de assinatura digital (Algoritmo de assinatura digital da curva elíptica, também conhecido como ECDSA)
- Os algoritmos hash (SHA-256 ou SHA-384)

Além disso, o padrão de IETF RFC 6460 especifica o Conjunto B compatível com perfis que definem a configuração detalhada do aplicativo e o comportamento necessário para estar em conformidade com o padrão do Conjunto B. Ele define dois perfis:

1. Um perfil compatível com o Conjunto B para uso com o TLS 1.2. Quando configurado para operação compatível com o Conjunto B, somente o conjunto restrito de algoritmos criptográficos listados são usados.
2. Um perfil de transição para uso com o TLS 1.0 ou o TLS 1.1. Este perfil permite a interoperabilidade com servidores não compatíveis com o Conjunto B. Quando configurado para a operação transicional do Conjunto B, a criptografia adicional e os algoritmos de hash podem ser usados.

O padrão do Conjunto B é conceitualmente semelhante ao FIPS 140-2, porque restringe o conjunto de algoritmos criptográficos ativados, a fim de fornecer um nível de garantia de segurança.

Em sistemas AIX, Linux, and Windows, o IBM MQ pode ser configurado para se adequar ao perfil TLS 1.2 compatível com o Conjunto B, mas não suporta o perfil de transição do Conjunto B. Veja informações adicionais na publicação “Criptografia do Conjunto B da NSA no IBM MQ” na página 43.

Referências relacionadas

“Federal Information Processing Standards” na página 23

O governo dos EUA produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. O National Institute for Standards and Technology (NIST) é um órgão importante ligado a sistemas e segurança de TI. O NIST produz recomendações e normas, inclusive Federal Information Processing Standards (FIPS).

Mecanismo de segurança do IBM MQ

Esta coleção de tópicos descreve mecanismos específicos no IBM MQ que implementam os vários conceitos de segurança

Protocolos de segurança TLS no IBM MQ

O IBM MQ suporta o protocolo da Segurança da Camada de Transporte (TLS) para fornecer segurança em nível de link para canais de mensagens e canais do MQI.

Os canais de mensagem e os canais de MQI podem usar o protocolo do TLS para fornecer a segurança em nível de link. Um MCA do responsável pela chamada é um cliente TLS e um MCA do respondente é um servidor do TLS.

O IBM MQ suporta as Versões 1.2 e 1.3 do protocolo TLS. As versões anteriores do TLS, bem como do SSL, não serão ativadas por padrão, mas poderão ser, se necessário. É possível especificar os algoritmos criptográficos que são usados pelo protocolo do TLS fornecendo um CipherSpec como parte da definição de canal.

Consulte “Ativando CipherSpecs” na página 427 para obter uma lista dos CipherSpecs suportados por IBM MQ e “CipherSpecs descontinuado” na página 443 para aqueles que foram descontinuados.

É possível usar os parâmetros SECPROT e SSLCIPH para exibir o protocolo de segurança e o CipherSpec em uso em um canal.

Em cada extremidade de um canal de mensagens, e na extremidade do servidor de um canal do MQI, o MCA atua em nome do gerenciador de filas ao qual está conectado. Durante o handshake do TLS, o MCA envia o certificado digital do gerenciador de filas para seu MCA parceiro na outra extremidade do canal. O código do IBM MQ na extremidade do cliente de um canal de MQI age em nome do usuário do aplicativo cliente IBM MQ. Durante o handshake do TLS, o código do IBM MQ envia o certificado digital do usuário ao MCA na extremidade do servidor do canal do MQI.

Os gerenciadores de filas e os usuários do cliente do IBM MQ não têm que ter certificados digitais pessoais associados a eles quando estiverem atuando como clientes TLS, a menos que SSLCAUTH(REQUIRED) seja especificado no lado do servidor do canal.

Os certificados digitais são armazenados em um *repositório de chaves*. O atributo **SSLKeyRepository** do gerenciador de filas especifica a localização do repositório de chaves que mantém o certificado digital do gerenciador de filas. Em um sistema do cliente de IBM MQ, a variável de ambiente MQSSLKEYR especifica a localização do repositório de chaves que mantém o certificado digital do usuário. Como alternativa, um aplicativo cliente do IBM MQ pode especificar seu local no campo **KeyRepository** da estrutura de opções de configuração de TLS, MQSCO, em uma chamada MQCONNX. Consulte os tópicos relacionados para obter mais informações sobre repositórios de chaves e como especificar onde eles estão localizados.

Suporte para TLS

O IBM MQ fornece suporte para TLS 1.2 e TLS 1.3 em todas as plataformas. Para obter mais informações sobre o protocolo do TLS, consulte as informações nos subtópicos.

Os clientes Java e JMS
Esses clientes usam a JVM para fornecer o suporte do TLS.

AIX, Linux, and Windows
O suporte do TLS é instalado com o IBM MQ.

IBM i
O suporte do TLS é integral para o sistema operacional do IBM i.

z/OS
O suporte do TLS é integral para o sistema operacional do z/OS. O suporte do TLS no z/OS é conhecido como *SSL do sistema*.

Para obter informações sobre quaisquer pré-requisitos para o suporte do TLS do IBM MQ, consulte Requisitos do sistema para IBM MQ.

Conceitos relacionados

"Protocolos de segurança criptográficos: TLS" na página 18
Os protocolos de criptografia fornecem conexões seguras, permitindo que dois grupos de comuniquem com privacidade e integridade de dados. O protocolo Transport Layer Security (TLS) surgiu desse do Secure Sockets Layer (SSL). O IBM MQ suporta o TLS.

O repositório de chaves SSL/TLS

Uma conexão TLS mutualmente autenticada requer um repositório de chaves em cada término da conexão. O repositório de chaves inclui certificados digitais e chaves privadas.

Estas informações usam o termo geral *repositório de chaves* para descrever o armazenamento para certificados digitais e suas chaves privadas associadas. O repositório de chaves é referido por diferentes nomes em diferentes plataformas e ambientes que suportam TLS:

- IBM i No IBM i: *armazenamento de certificados*
- No Java e no JMS: *keystore* e *armazenamento confiável*
- ALW No AIX, Linux, and Windows: *Arquivo do banco de dados de chave*
- z/OS No z/OS: *conjunto de chaves*

Para obter mais informações, veja "Certificados Digitais" na página 13 e "Conceitos de TLS (Transport Layer Security)" na página 18.

Uma conexão TLS mutualmente autenticada requer um repositório de chaves em cada término da conexão. O repositório de chaves pode conter os seguintes certificados e solicitações:

- Vários certificados de CA de várias Autoridades de Certificação que permitem que o gerenciador de filas ou o cliente verifique certificados recebidos de seu parceiro na extremidade remota da conexão. Certificados individuais podem estar em uma cadeia de certificados.
- Um ou mais certificados pessoais recebidos de uma Autoridade de Certificação. É possível associar um certificado pessoal diferente a cada gerenciador de filas ou IBM MQ MQI client. Os certificados pessoais serão essenciais em um cliente TLS se a autenticação mútua for necessária. Se a autenticação mútua não for necessária, os certificados pessoais não serão necessários no cliente. O repositório de chaves também pode conter a chave privada correspondente a cada certificado pessoal.
- As solicitações de certificados que estão aguardando para serem assinadas por uma autoridade de certificação confiável.

Para obter mais informações sobre como proteger seu repositório de chaves, consulte "Protegendo repositórios de chaves do IBM MQ" na página 27.

A localização do repositório de chaves depende da plataforma que você está utilizando:

IBM i **IBM i**
O repositório de chaves é um armazenamento de certificados. O armazenamento de certificados padrão do sistema se localiza em `/QIBM/UserData/ICSS/Cert/Server/Default` no IFS (integrated file system). O IBM MQ armazena a senha para o armazenamento de certificados em um *arquivo stash de senha*. Por exemplo, o arquivo stash para o gerenciador de filas `QM1` é `/QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth`.

Como alternativa, é possível especificar que o armazenamento de certificados do sistema IBM i deve ser usado em seu lugar. Para fazer isso, mude o valor do atributo **SSLKEYR** do gerenciador

de filas para *SYSTEM. Esse valor indica que o gerenciador de filas deve usar o armazenamento de certificados do sistema e que o gerenciador de filas seja registrado para ser usado como um aplicativo com o Digital Certificate Manager (DCM).

O armazenamento de certificados também contém a chave privada para o gerenciador de filas.

ALW **Sistemas AIX, Linux, and Windows**

O repositório de chaves é um arquivo do banco de dados de chaves. Por exemplo, no AIX and Linux, o arquivo do banco de dados de chaves padrão para o QM1 do gerenciador de filas é `/var/mqm/qmgrs/QM1/ssl/key.kdb`. Se IBM MQ estiver instalado no local padrão, o caminho equivalente no Windows será `C:\ProgramData \IBM \MQ\Qmgrs\QM1\ssl\key.kdb`.

Para acessar um arquivo de banco de dados de chave, IBM MQ deve ser fornecido com a senha para o banco de dados de chave Isso pode ser feito diretamente ou por meio de um arquivo stash de senha Se um arquivo stash de senha for usado, ele deverá estar no mesmo diretório e ter a mesma raiz de arquivo que o banco de dados de chave e deverá terminar com o sufixo `.sth`, por exemplo `/var/mqm/qmgrs/QM1/ssl/key.sth`

Nota: As placas de hardware de criptografia PKCS #11 podem conter os certificados e chaves que são, em outras situações, armazenados em um arquivo do banco de dados de chaves. Quando os certificados e chaves são mantidos em placas PKCS #11, o IBM MQ ainda requer acesso a um arquivo do banco de dados de chaves e um arquivo stash de senha.

Em sistemas AIX, Linux, and Windows, o banco de dados de chaves também contém a chave privada para o certificado pessoal associado ao gerenciador de filas ou IBM MQ MQI client.

z/OS **z/OS**

Os certificados são mantidos em um conjunto de chaves em z/OS.

Outros gerenciadores de segurança externa (ESMs) também usam conjuntos de chave para armazenar certificados.

As chaves privadas são gerenciadas pelo RACF.

Protegendo repositórios de chaves do IBM MQ

O repositório de chaves para o IBM MQ é um arquivo. Certifique-se de que somente o usuário pretendido possa acessar o arquivo repositório de chaves. Isso impede um intruso ou outro usuário não autorizado de copiar o arquivo repositório de chaves para outro sistema, e então configurar um ID de usuário idêntico naquele sistema para personificar o usuário pretendido.

As permissões sobre os arquivos dependem da umask do usuário e da ferramenta usada. No Windows, as contas do IBM MQ requerem permissão `BypassTraverseChecking`, que significa que as permissões das pastas no caminho de arquivo não têm efeito.

Verifique as permissões dos arquivos de repositório de chaves e certifique-se de que os arquivos e pasta recipiente não sejam globalmente legíveis, de preferência, nem legíveis por grupos.

Seja qual for o sistema usado, é uma boa prática tornar o keystore somente leitura, apenas com o administrador tendo permissão de ativar operações de gravação para executar manutenção.

Na prática, você deve proteger todos os keystores, seja qual for a localização e se forem protegidos por senha ou não; proteja os repositórios de chaves.

Rótulos de Certificados Digitais, Entendendo os Requisitos

Ao configurar o TLS para usar certificados digitais, pode haver requisitos de rótulo específicos que devem ser seguidos, dependendo da plataforma usada e do método usado para a conexão.

O que é rótulo certificado?

Um rótulo de certificado é um identificador exclusivo que representa um certificado digital armazenado em um repositório de chaves e que fornece um nome legível conveniente com o qual se referir a um determinado certificado ao executar funções de gerenciamento de chaves. Você designa o rótulo certificado ao incluir um certificado em um repositório de chaves pela primeira vez.

O rótulo do certificado é separado dos campos **Subject Distinguished Name** ou **Subject Common Name** do certificado Observe que **Subject Distinguished Name** e **Subject Common Name** são campos dentro do próprio certificado. Eles são definidos quando o certificado é criado e não pode ser alterado. Se necessário, entretanto, é possível mudar o rótulo associado a um certificado digital.

Sintaxe do rótulo certificado

Um rótulo certificado pode conter letras, números e pontuação com as condições a seguir:

- Multi O rótulo certificado pode conter até 64 caracteres.
- z/OS O rótulo certificado pode conter até 32 caracteres.
- O rótulo do certificado pode conter espaços
- Os rótulos fazem distinção entre maiúsculas e minúsculas.
- Nos sistemas que usam EBCDIC katakana, não é possível usar caracteres minúsculos.

Requisitos adicionais para os valores do rótulo certificado são especificados nas seções a seguir.

Como o rótulo certificado é usado?

O IBM MQ usa rótulos certificados para localizar um certificado pessoal que é enviado durante o handshake TLS. Isso elimina a ambiguidade quando existe mais de um certificado pessoal no repositório de chaves.

É possível configurar o rótulo certificado para um valor de sua escolha. Se você não configurar um valor, um rótulo padrão será usado, seguindo uma convenção de nomenclatura dependendo da plataforma que você está usando. Para obter detalhes, consulte as seções a seguir sobre plataformas específicas.

Notas:

1. Não é possível configurar o rótulo certificado você mesmo em sistemas Java ou JMS.
2. Os canais autodefinidos criados por uma saída de channel automatic definition (CHAD) não podem configurar o rótulo certificado, porque o handshake TLS ocorreu pelo tempo que o canal é criado. Configurar o rótulo certificado em uma saída CHAD para canais de entrada não terá efeito.

Neste contexto, um cliente TLS se refere ao parceiro de conexão que inicia o handshake, que pode ser um cliente IBM MQ ou outro gerenciador de filas.

Durante o handshake do TLS, o cliente do TLS sempre obtém e valida um certificado digital a partir do servidor. Com a implementação do IBM MQ, o servidor TLS sempre solicita um certificado do cliente e o cliente sempre fornece um certificado para o servidor, se for encontrado. Se o cliente não puder localizar um certificado pessoal, o cliente enviará uma resposta no `certificate` para o servidor

O servidor do TLS sempre valida o certificado de cliente, se um for enviado. Se o cliente não enviar um certificado, a autenticação falhará se a extremidade do canal que está agindo como o servidor TLS estiver definida com o parâmetro **SSLCAUTH** configurado como *REQUIRED* ou um valor de parâmetro **SSLPEER** configurado.

Observe que os canais de entrada (incluindo receptor, solicitante, receptor de cluster, servidor não qualificado e canais de conexão do servidor) enviarão o certificado configurado apenas se a versão do IBM MQ do peer remoto suportar totalmente a configuração do rótulo certificado e o canal estiver usando um CipherSpec de TLS.

Um canal do servidor não qualificado é aquele que não tem o campo CONNAME configurado.

Em todos os outros casos, parâmetro **CERTLABL** do gerenciador de filas determina o certificado enviado. Em particular, o seguinte somente sempre recebe o certificado configurado pelo parâmetro **CERTLABL** do gerenciador de filas, independentemente da configuração do rótulo do canal específico:

- Os clientes Java e JMS que suportam a Indicação de nome do servidor (SNI), ou seja, certificados em uma base canal por canal.
- Versões do IBM MQ anteriores à IBM MQ 8.0.

- Clientes .NET gerenciados

Além disso, o certificado usado por um canal deve ser apropriado para o canal CipherSpec - consulte “Certificados digitais e compatibilidade de CipherSpec no IBM MQ” na página 48 para obter mais informações.

O IBM MQ 8.0 e mais recente suportam o uso de vários certificados no mesmo Gerenciador de Filas, utilizando um rótulo de certificado por canal especificado por meio do atributo **CERTLABL** na definição de canal. Os canais de entrada para o gerenciador de filas (por exemplo, conexão ou receptor do servidor) dependem de detectar o nome do canal usando Name Server Indication (SNI) TLS, a fim de apresentar o certificado correto do gerenciador de filas. Para obter mais informações sobre o uso de vários certificados em um Gerenciador de Filas, consulte “Como o IBM MQ fornece o recurso de vários certificados” na página 30.

Se um canal se conectar ao gerenciador de filas de destino por meio do IBM MQ Internet Pass-Thru (MQIPT) e a rota MQIPT tiver **SSLServer** e **SSLClient** configurado, haverá duas sessões TLS separadas entre os terminais. O MQIPT pode ser configurado para permitir que vários certificados sejam usados pelo gerenciador de filas de destino, configurando o SNI para o nome do canal ou passando pelo SNI recebido na conexão de entrada para a rota Para obter mais informações sobre o suporte a vários certificados e MQIPT, consulte Suporte a vários certificados do IBM MQ com o MQIPT.

Para obter mais informações sobre como conectar um gerenciador de filas usando autenticação unilateral, ou seja, quando o cliente TLS não envia um certificado, veja Conectando dois gerenciadores de filas usando autenticação unilateral.

Sistemas multiplataformas

Multi

Em Multiplataformas, o servidor TLS envia um certificado para o cliente.

Para gerenciadores de filas e clientes respectivamente, as fontes a seguir são procuradas em sequência para um valor não vazio. O primeiro valor não vazio determina o rótulo certificado. O rótulo certificado deve existir no repositório de chaves. Se não for localizado um certificado correspondente nas maiúsculas e minúsculas e no formato corretos que corresponda a um rótulo, ocorrerá um erro e o handshake TLS falhará.

Gerenciadores de filas

1. Atributo **CERTLABL** do rótulo certificado do canal.
2. Atributo **CERTLABL** do rótulo certificado do gerenciador de filas.
3. Um padrão, que está no formato: `ibmwebspheremq` com o nome do gerenciador de filas anexado, tudo em minúsculas. Por exemplo, para um gerenciador de filas chamado `QM1`, o rótulo certificado padrão é `ibmwebspheremqqm1`.

Clientes do IBM MQ

1. Atributo de rótulo de certificado **CERTLABL** na definição de canal CLNTCONN.
2. Atributo **`CertificateLabel`** da estrutura MQSCO.
3. Variável de ambiente **MQCERTLABL**.
4. Arquivo `.ini` de cliente (em sua seção SSL) atributo **`CertificateLabel`**
5. Um padrão, que está no formato: `ibmwebspheremq` com o ID do usuário que o aplicativo cliente está executando como anexado, tudo em minúsculas. Por exemplo, para um ID do usuário de USER1, o rótulo certificado padrão é `ibmwebspheremquser1`.

Sistemas z/OS

z/OS

Clientes IBM MQ não são suportados no z/OS. No entanto, um gerenciador de filas do z/OS pode agir na função de um cliente TLS ao iniciar uma conexão ou um servidor TLS ao aceitar uma solicitação de

conexão. Os requisitos de rótulo certificado para gerenciadores de filas do z/OS se aplicam a ambas as funções e são diferentes dos requisitos em Multiplataformas.

Para gerenciadores de filas e clientes respectivamente, as fontes a seguir são procuradas em sequência para um valor não vazio. O primeiro valor não vazio determina o rótulo certificado. O rótulo certificado deve existir no repositório de chaves. Se não for localizado um certificado correspondente nas maiúsculas e minúsculas e no formato corretos que corresponda a um rótulo, ocorrerá um erro e o handshake TLS falhará.

1. Atributo de rótulo certificado do canal, **CERTLABL**.
2. Se compartilhado, o atributo do rótulo certificado do grupo de filas compartilhadas, **CERTQSGL**.

   Se não compartilhado, o atributo do rótulo certificado do gerenciador de filas, **CERTLABL**.
3. Um padrão, que está no formato: `ibmWebSphereMQ` com o nome do gerenciador de filas ou o grupo de filas compartilhadas anexado. Observe que esta sequência faz distinção entre maiúsculas e minúsculas deve ser escrita conforme mostrado. Por exemplo, para um gerenciador de filas chamado `QM1`, o rótulo certificado padrão é `ibmWebSphereMQQM1`.
4. Se não houver um certificado localizado com o formato na opção “3” na página 30, o IBM MQ tentará utilizar o certificado marcado como padrão no conjunto de chaves.

Para obter informações sobre como exibir o repositório de chaves, consulte “Locating the key repository for a queue manager on z/OS” na página 315.

Os clientes IBM MQ Java e IBM MQ JMS

Os clientes IBM MQ Java e IBM MQ JMS usam os recursos de seu provedor de Java Secure Socket Extension (JSSE) para selecionar um certificado pessoal durante o handshake TLS e, portanto, não estão sujeitos a requisitos de rótulo certificado.

O comportamento padrão é que o cliente JSSE itere através de certificados no repositório de chaves, selecionando o primeiro certificado pessoal aceitável localizado. No entanto, esse comportamento é apenas um padrão, e depende da implementação do provedor JSSE.

Além disso, a interface JSSE é altamente customizável por meio de configuração e acesso direto no tempo de execução pelo aplicativo. Consulte a documentação fornecida pelo provedor JSSE para obter detalhes específicos.

Para resolução de problemas ou para entender melhor o handshake executado pelo aplicativo cliente do IBM MQ Java em combinação com o provedor JSSE específico, é possível ativar a depuração configurando `javax.net.debug=ssl` no ambiente JVM.

É possível configurar a variável dentro do aplicativo, através da configuração ou entrando em `-Djavax.net.debug=ssl` na linha de comandos.

Linux *Como o IBM MQ fornece o recurso de vários certificados*

A Indicação de Nome de Servidor (SNI) é uma extensão do protocolo TLS que permite que um cliente indique qual serviço ele requer. Na terminologia do IBM MQ isso equivale a um canal.

A extensão SNI é usada por IBM MQ para permitir que vários certificados sejam especificados em diferentes canais usando o parâmetro CERTLABL na definição do canal.

O endereço do SNI usado por IBM MQ é baseado no nome do canal que está sendo solicitado, seguido por um sufixo de `.chl.mq.ibm.com`.

Os nomes de canais do IBM MQ são mapeados para serem nomes de SNI válidos da maneira a seguir:

- As letras maiúsculas de A a Z são convertidas em letras minúsculas
- Os dígitos de `0` a `9` não são mudados
- Todos os outros caracteres, incluindo letras minúsculas a a `z`, são convertidos em seu código de caracteres ASCII hexadecimal de dois dígitos (em letras minúsculas), seguido por um hífen.
  - As letras minúsculas de a a `z` são mapeadas para o hexadecimal `61-` a `7a-`, respectivamente

- o símbolo de porcentagem (%) é mapeado para o hexadecimal `25-`
- o hífen (-) é mapeado para o hexadecimal `2d-`
- o ponto (.) é mapeado para o hexadecimal `2e-`
- a barra (/) é mapeada para o hexadecimal `2f-`
- o sublinhado (_) é mapeado para o hexadecimal `5f-`

Nas plataformas EBCDIC, o nome de canal é convertido em ASCII antes que esse mapeamento seja aplicado.

Como exemplo, o nome do canal `TO.QMGR1` é mapeado para um endereço SNI de `to2e-qmgr1.chl.mq.ibm.com`.

Em contraste, o nome do canal em minúscula `to.qmgr1` é mapeado para o endereço SNI de `74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com`.

Nota: Em ambientes em que a URL SNI gerada deve se adequar às especificações de formatação de URL, por exemplo quando um cliente está se conectando a um Gerenciador de Filas em execução no Red Hat® OpenShift® por meio de uma rota do Red Hat OpenShift, o nome do canal não deve terminar com uma letra minúscula.

A propriedade **OutboundSNI** da sub-rotina SSL permite que você selecione se o SNI deve ser configurado com o nome do canal de destino IBM MQ para o sistema remoto ao iniciar uma conexão TLS, ou para o nome do host. Para obter informações adicionais sobre a propriedade **OutboundSNI**, consulte Sub-rotina SSL do arquivo qm.ini e Sub-rotina SSL do arquivo de configuração do cliente.

Vários certificados requerem que o SNI seja configurado para o nome do canal IBM MQ Se um nome do host, customizado ou nenhum SNI for usado para se conectar a um canal IBM MQ com um rótulo certificado configurado, o aplicativo de conexão será rejeitado com um MQRC_SSL_INITIALIZATION_ERROR e uma mensagem AMQ9673 será impressa nos logs de erros do gerenciador de filas remotas.

Se um canal se conectar ao Gerenciador de Filas de destino por meio do IBM MQ Internet Pass-Thru (MQIPT), o MQIPT deverá ser configurado para definir o SNI para o nome do canal ou para transmitir o SNI recebido na conexão de entrada para a rota, para permitir que vários certificados sejam utilizados pelo Gerenciador de Filas de destino. Para obter mais informações sobre o suporte a vários certificados e MQIPT, consulte Suporte a vários certificados do IBM MQ com o MQIPT.

Para obter mais informações sobre como essa propriedade é usada, consulte Conectando-se a um gerenciador de filas implementado em um cluster Red Hat OpenShift.

Atualizando o repositório de chaves do gerenciador de filas
Ao mudar o conteúdo de um repositório de chaves, os processos do gerenciador de filas existentes não selecionam o novo conteúdo até que um comando REFRESH SECURITY TYPE (SSL) seja emitido ou o gerenciador de filas seja reiniciado.

Para obter mais informações sobre o comando REFRESH SECURITY TYPE(SSL), consulte REFRESH SECURITY.

Se o gerenciador de filas criar um novo processo de canal (usando `amqrmppa` ou **runmqchl**) após alterar o conteúdo do keystore, o novo processo começará a usar os novos certificados imediatamente, enquanto os processos existentes continuarão a usar sua cópia em cache do keystore. Consulte "Quando as mudanças nos certificados ou no repositório de chaves tornam-se efetivas no AIX, Linux, and Windows" na página 311 para obter mais detalhes.

Observe que vários canais em execução poderiam estar usando versões diferentes do repositório de chaves até você emitir um comando REFRESH SECURITY TYPE (SSL).

Também é possível atualizar um repositório de chaves usando os comandos PCF ou o IBM MQ Explorer. Para obter mais informações, veja o comando MQCMD_REFRESH_SECURITY e o tópico *Atualizando a segurança TLS (Segurança da Camada de Transporte)* na seção do IBM MQ Explorer da documentação deste produto.

Conceitos relacionados

“Atualizando a visualização de um cliente do conteúdo do repositório de chaves SSL/TLS e configurações de SSl/TLS” na página 32
Para atualizar o aplicativo cliente com o conteúdo atualizado do repositório de chaves, deve-se parar e reiniciar o aplicativo cliente.

Atualizando a visualização de um cliente do conteúdo do repositório de chaves SSL/TLS e configurações de SSl/TLS

Para atualizar o aplicativo cliente com o conteúdo atualizado do repositório de chaves, deve-se parar e reiniciar o aplicativo cliente.

Não é possível atualizar a segurança em um cliente IBM MQ; não há equivalente do comando REFRESH SECURITY TYPE(SSL) para clientes (consulte REFRESH SECURITY) para obter mais informações.

Deve-se parar e reiniciar o aplicativo sempre que o certificado de segurança for mudado, para atualizar o aplicativo cliente com o conteúdo atualizado do repositório de chaves.

Se reiniciar o canal atualiza as configurações, e se o seu aplicativo tem reconexão lógica, é possível atualizar a segurança no cliente emitindo o comando STOP CHL STATUS(INACTIVE).

Conceitos relacionados

“Atualizando o repositório de chaves do gerenciador de filas” na página 31
Ao mudar o conteúdo de um repositório de chaves, os processos do gerenciador de filas existentes não selecionam o novo conteúdo até que um comando REFRESH SECURITY TYPE (SSL) seja emitido ou o gerenciador de filas seja reiniciado.

Proteção de senha do MQCSP

As credenciais de autenticação que são especificadas na estrutura MQCSP podem ser protegidas usando o recurso de proteção de senha do MQCSP IBM MQ ou criptografadas usando a criptografia TLS.

Os aplicativos IBM MQ client podem fornecer um ID do usuário e uma senha quando se conectam a um gerenciador de filas. V 9.4.0 De IBM MQ 9.4.0, os aplicativos também podem fornecer um token de autenticação como um método alternativo de autenticação. Essas credenciais são enviadas ao gerenciador de fila em uma estrutura MQCSP.

Se o canal estiver usando criptografia TLS, as credenciais no MQCSP serão criptografadas de acordo com a especificação de cifra TLS. Se o canal não estiver usando a criptografia TLS, o IBM MQ poderá proteger essas credenciais antes que elas sejam enviadas pela rede, para evitar o envio de credenciais por uma rede em texto sem formatação. O recurso IBM MQ que protege essas credenciais é chamado proteção de senha MQCSP.

Se a proteção de senha MQCSP for usada, os dados a seguir na estrutura MQCSP serão protegidos:

- A senha, se o campo `MQCSP.AuthenticationType` for configurado como `MQCSP_AUTH_USER_ID_AND_PW`.
- V 9.4.0 O token de autenticação, se o campo `MQCSP.AuthenticationType` for configurado para `MQCSP_AUTH_ID_TOKEN`

Importante: A proteção de senha MQCSP é útil para propósitos de teste e desenvolvimento, já que usar a proteção de senha MQCSP é mais simples do que configurar a criptografia TLS, mas não tão seguro. Para propósitos de produção, use a criptografia TLS em preferência à proteção de senha do IBM MQ , especialmente quando a rede entre o cliente e o gerenciador de filas não for confiável, já que a criptografia TLS é mais segura

Se você estiver preocupado com qual criptografia está sendo usada e quanta proteção ela oferece, será necessário usar a criptografia TLS completa. Com TLS, os algoritmos são publicamente conhecidos e é possível selecionar o apropriado para sua empresa usando o atributo do canal **SSLCIPH** .

Para obter mais informações sobre a estrutura MQCSP, consulte Estrutura MQCSP.

As credenciais na estrutura MQCSP são protegidas usando a proteção de senha IBM MQ se todas as seguintes condições forem atendidas:

- Ambas as extremidades da conexão estão usando o IBM MQ 8.0 ou mais recente.
- O canal não está usando a criptografia TLS. Um canal não está usando a criptografia TLS se o canal tiver um atributo **SSLCIPH** em branco ou se o atributo **SSLCIPH** estiver configurado para uma especificação de cifra que não fornece criptografia. Cifras nulas, por exemplo, NULL_SHA, não fornecem criptografia.
- O campo `MQCSP.AuthenticationType` é configurado como `MQCSP_AUTH_USER_ID_AND_PWD` ou `MQCSP_AUTH`. Para obter mais informações sobre o campo `MQCSP.AuthenticationType` , consulte **`AuthenticationType`**..
- Se o cliente for IBM MQ Explorer e o modo de compatibilidade de identificação do usuário não estiver ativado, Esse modo não é o modo padrão usado pelo IBM MQ Explorer para enviar um ID do usuário e uma senha. Essa condição é aplicável apenas a IBM MQ Explorer

Se alguma dessas condições não for atendida, as credenciais não serão protegidas com a proteção de senha MQCSP. Se o valor do atributo **`PasswordProtection`** proibir que as credenciais sejam enviadas em texto simples e o canal não estiver usando a criptografia TLS, a conexão falhará e um código de razão MQRC_PASSWORD_PROTECTION_ERROR (2594) será retornado.

A definição de configuração `PasswordProtection`

O atributo **`PasswordProtection`** na sub-rotina **`Channels`** dos arquivos de configuração do cliente e do gerenciador de filas pode evitar que as credenciais sejam enviadas em texto simples.

Nota: Esse atributo é relevante apenas para conexões que não usam a criptografia TLS (TLS). As credenciais são criptografadas usando TLS em vez de serem protegidas com a proteção de senha MQCSP se a conexão usar a criptografia TLS.

O atributo pode ser configurado para um dos valores a seguir: O valor padrão é `compatible`.

compatíveis
As credenciais serão enviadas em texto simples se o gerenciador de filas ou o cliente estiver executando uma versão de IBM MQ anterior a IBM MQ 8.0. Ou seja, as credenciais podem ser enviadas por meio de uma rede em texto simples para compatibilidade com versões do IBM MQ que não suportam a proteção de senha do MQCSP.

As credenciais serão protegidas pela proteção de senha MQCSP se o gerenciador de filas e o cliente estiverem executando uma versão de IBM MQ em IBM MQ 8.0 ou posterior.

A conexão falha antes de as credenciais serem enviadas se o gerenciador de filas e o cliente estiverem executando uma versão de IBM MQ em IBM MQ 8.0 ou posterior e o campo `MQCSP.AuthenticationType` não estiver configurado como `MQCSP_AUTH_USER_ID_AND_PW` ou `MQCSP_AUTH_ID_TOKEN`.

sempre
As credenciais não devem ser enviadas por uma rede desprotegida.

As credenciais serão protegidas pela proteção de senha MQCSP se o gerenciador de filas e o cliente estiverem executando uma versão de IBM MQ em IBM MQ 8.0 ou posterior.

A conexão falha antes que as credenciais sejam enviadas nos casos a seguir:

- O campo `MQCSP.AuthenticationType` não está configurado como `MQCSP_AUTH_USER_ID_AND_PW` ou `MQCSP_AUTH_ID_TOKEN`
- O gerenciador de filas ou o cliente está executando uma versão de IBM MQ anterior a IBM MQ 8.0

opcional
As credenciais serão protegidas pela proteção de senha do MQCSP se o gerenciador de fila e o cliente estiverem executando uma versão de IBM MQ em IBM MQ 8.0 ou mais recente e o campo `MQCSP.AuthenticationType` for configurado como `MQCSP_AUTH_USER_ID_AND_PW` ou `MQCSP_AUTH_ID_TOKEN` Caso contrário, as credenciais serão enviadas em texto simples.

avisar
Qualquer cliente tem permissão para enviar credenciais de texto simples. Se credenciais de texto simples forem recebidas, a mensagem de aviso AMQ9297W será gravada nos logs de erro do gerenciador de filas.

Essa opção pode ser especificada apenas no arquivo de configuração do gerenciador de filas

Para clientes Java e JMS , o comportamento do atributo **PasswordProtection** muda dependendo se o cliente usa o modo de compatibilidade ou o modo MQCSP:

- Se clientes Java e JMS estiverem operando no modo de compatibilidade, uma estrutura MQCSP não será usada para enviar o ID do usuário e a senha quando o cliente se conectar. Portanto, o comportamento do atributo **PasswordProtection** é igual ao comportamento descrito para clientes que estão executando uma versão de IBM MQ anterior a IBM MQ 8.0.
- Se os clientes Java e JMS estiverem operando no modo MQCSP, o comportamento do atributo **PasswordProtection** será o comportamento descrito.

Para obter mais informações sobre autenticação de conexão com clientes Java e JMS , consulte "Autenticação de conexão com o cliente Java" na página 86.

Proteção de senha do MQCSP e MQIPT

V 9.4.0

Se um cliente se conectar a um gerenciador de filas por IBM MQ Internet Pass-Thru (MQIPT), a rota MQIPT poderá ser configurada para incluir ou remover a criptografia TLS. Ou seja, a rota MQIPT pode ser configurada com SSLServer=true e SSLClient=falseou com SSLServer=true e SSLClient=false Nessa situação, o cliente e o gerenciador de filas podem falhar ao concordar com um algoritmo de proteção de senha, pois uma extremidade do canal está usando a criptografia TLS e a outra não. Isso faz a conexão falhar com o código de razão MQRC_PASSWORD_PROTECTION_ERROR (2594).

A partir do IBM MQ 9.4.0, o MQIPT pode incluir ou remover a proteção para credenciais em estruturas MQCSP, para manter a compatibilidade entre o cliente e o gerenciador de filas para rotas do MQIPT que incluem ou removem a criptografia TLS. A proteção de senha do MQCSP no MQIPT é configurada usando a propriedade de rota **PasswordProtection**

O valor padrão da propriedade **PasswordProtection** é obrigatório. Esse valor significa que o MQIPT pode incluir, mas não remover, a proteção de senha do MQCSP. As conexões com uma rota do MQIPT que inclui a criptografia TLS podem falhar com o código de razão MQRC_PASSWORD_PROTECTION_ERROR (2594) com esse valor de **PasswordProtection**. Para resolver esse problema, configure o valor da propriedade **PasswordProtection** para compatível na configuração de rota do MQIPT

Para obter mais informações sobre a propriedade **PasswordProtection** em MQIPT, consulte PasswordProtection

Digital Certificate Manager (DCM)

Use o DCM para gerenciar certificados digitais e chaves privadas em IBM i.

O Digital Certificate Manager (DCM) permite gerenciar certificados digitais e usá-los em aplicativos seguros no servidor IBM i. Com o Digital Certificate Manager, é possível solicitar e processar certificados digitais das Autoridades de Certificação (CAs) ou de terceiros. Também é possível atuar como uma Autoridade de Certificação local para criar e gerenciar certificados digitais para seus usuários.

O DCM também suporta o uso de Listas de Revogação de Certificado para fornecer um certificado e um processo de validação de aplicativo mais consistentes. É possível usar o DCM para definir o local onde uma CRL de Autoridade de Certificação específica reside em um servidor LDAP, para que o IBM MQ possa verificar se um certificado específico ainda não foi revogado.

O DCM suporta e pode detectar automaticamente certificados em uma variedade de formatos. Quando o DCM detectar um certificado codificado pelo PKCS #12, ou um certificado PKCS #7 que contém dados criptografados, ele automaticamente solicitará ao usuário inserir a senha que foi utilizada para criptografar o certificado. O DCM não solicita os certificados PKCS #7 que não contém dados criptografados.

O DCM fornece uma interface com o usuário baseada em navegador que pode ser utilizada para gerenciar certificados digitais para os aplicativos e usuários. A interface com o usuário é dividida em dois quadros principais: um quadro de navegação e um quadro de tarefas.

Utilize o quadro de navegação para selecionar as tarefas para gerenciar os certificados ou os aplicativos que os utilizam. Algumas tarefas individuais são mostradas diretamente no quadro de navegação principal, mas a maioria das tarefas no quadro de navegação são organizadas em categorias. Por exemplo, Gerenciar Certificados é uma categoria de tarefa que contém várias tarefas individuais orientadas, tais como Visualizar Certificado, Renovar Certificado e Importar Certificado. Se um item no quadro de navegação for uma categoria que contém mais de uma tarefa, uma seta é exibida à esquerda dele. A seta indica que quando um link da categoria é selecionado, uma lista expandida de tarefas será exibida, permitindo escolher quais tarefas serão desempenhadas.

Para obter informações importantes sobre DCM, consulte as seguintes publicações do IBM Redbooks:

- Segurança de rede com fio do *IBM i: OS/400 V5R1 DCM e aprimoramentos criptográficos*, SG24-6168. Especificamente, consulte os apêndices para informações essenciais sobre como configurar seu sistema IBM i como um CA local.
- *segurança de internet AS/400: desenvolvendo uma infraestrutura de certificado digital*, SG24-5659. Especificamente, consulte o Capítulo 5. Gerenciador de certificado *digital para AS/400*, que explica o AS/400 DCM.

FIPS (Federal Information Processing Standards)

Este tópico apresenta o Federal Information Processing Standards (FIPS) Cryptomodule Validation Program do National Institute of Standards and Technology dos EUA e as funções criptográficas que podem ser usadas nos canais TLS.

Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o IBM Crypto for C (ICC) certificado e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na NIST CMVP modules in process list.

O IBM MQ Operator 3.2.0 e a imagem do contêiner do gerenciador de filas 9.4.0.0 em diante são baseados no UBI 9 A conformidade do FIPS 140-3 está pendente atualmente e seu status pode ser visualizado procurando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" no NIST CMVP modules in process list.

Estas informações se aplicam às seguintes plataformas:

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

ALW Para obter mais informações sobre a conformidade com o FIPS 140-2 de uma conexão TLS do IBM MQ no AIX, Linux, and Windows, consulte “Federal Information Processing Standards (FIPS) para AIX, Linux, and Windows” na página 36.

z/OS Para obter mais informações sobre a conformidade com o FIPS 140-2 de uma conexão TLS do IBM MQ no z/OS, consulte “Federal Information Processing Standards (FIPS) for z/OS” na página 39.

Se o hardware de criptografia estiver presente, os módulos de criptografia usados pelo IBM MQ podem ser configurados para ser aqueles fornecidos pelo fabricante do hardware. Se isso for feito, a configuração só ficará em conformidade com o FIPS se esses módulos criptográficos forem certificados pelo FIPS.

Com o passar do tempo, os Federal Information Processing Standards são atualizados para refletirem novos ataques contra algoritmos de criptografia e protocolos. Por exemplo, alguns CipherSpecs podem deixar de ser certificados por FIPS. Quando tais mudanças ocorrem, o IBM MQ também é atualizado para implementar o padrão mais recente. Como um resultado, você poderá ver as mudanças no comportamento após a aplicação da manutenção.

Conceitos relacionados

“Especificando que Apenas CipherSpecs Certificados por FIPS São Usados no Tempo de Execução no Cliente de MQI” na página 273

Crie seus repositórios de chaves usando software compatível com FIPS e, em seguida, especifique que o canal deve usar CipherSpecs certificados por FIPS.

Tarefas relacionadas

Ativando o TLS no IBM MQ classes for Java

Usando Segurança da Camada de Transporte (TLS) com o IBM MQ classes for JMS

Referências relacionadas

Propriedades de TLS de objetos do JMS

“Comandos runmqakm e runmqktool em AIX, Linux, and Windows” na página 550

Em sistemas AIX, Linux, and Windows , use os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool) para gerenciar chaves e certificados..

“Federal Information Processing Standards” na página 23

O governo dos EUA produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. O National Institute for Standards and Technology (NIST) é um órgão importante ligado a sistemas e segurança de TI. O NIST produz recomendações e normas, inclusive Federal Information Processing Standards (FIPS).

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

Quando a criptografia é necessária em um canal SSL/TLS em AIX, Linux, and Windows sistemas, IBM MQ usa um pacote de criptografia chamado IBM Crypto for C (ICC). Nas plataformas do AIX, Linux, and Windows , o software ICC passou no Programa de Validação de Criptomódulos do Federal Information Processing Standards (FIPS) do US National Institute of Standards and Technology, no nível 140-2.

Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o IBM Crypto for C (ICC) certificado e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na NIST CMVP modules in process list.

O IBM MQ Operator 3.2.0 e a imagem do contêiner do gerenciador de filas 9.4.0.0 em diante são baseados no UBI 9 A conformidade do FIPS 140-3 está pendente atualmente e seu status pode ser visualizado procurando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" no NIST CMVP modules in process list.

A conformidade com o FIPS 140-2 de uma conexão TLS do IBM MQ em sistemas AIX, Linux, and Windows é conforme a seguir:

- Para todos os canais de mensagens do IBM MQ (exceto os tipos de canais CLNTCONN), a conexão será compatível com o FIPS se as seguintes condições forem atendidas:
  - A versão instalada do IBM Global Security Kit (GSKit) ICC foi certificada compatível com FIPS 140-2 na versão do sistema operacional e arquitetura de hardware instalados.
  - O atributo SSLFIPS do gerenciador de filas foi configurado para YES.
  - Todos os repositórios de chaves foram criados e manipulados usando somente software compatível com FIPS, como **runmqakm** com a opção `-fips`.
  - O acesso a todos os repositórios de chaves é fornecido usando um arquivo stash e não o atributo **KEYRPWD** do Gerenciador de Filas
- Para todos os aplicativos IBM MQ MQI client , a conexão usa GSKit e será compatível com FIPS se as condições a seguir forem atendidas:
  - A versão instalada do GSKit ICC foi certificada compatível com FIPS 140-2 na versão do sistema operacional e arquitetura de hardware instalados.
  - Você especificou que somente a criptografia certificada para FIPS deve ser usada, conforme descrito no tópico relacionado para o cliente de MQI.
  - Todos os repositórios de chaves foram criados e manipulados usando somente software compatível com FIPS, como **runmqakm** com a opção `-fips`.

  - O acesso a todos os repositórios de chave é fornecido usando um arquivo stash e não o mecanismo de senha do repositório de chave
- Para aplicativos IBM MQ classes for Java que usam o modo cliente, a conexão usa as implementações TLS do JRE e será compatível com o FIPS se as condições forem atendidas:
  - O Java Runtime Environment usado para executar o aplicativo for compatível com o FIPS na versão do sistema operacional e arquitetura de hardware instaladas.
  - Você especificou que somente a criptografia certificada para FIPS deve ser usada, conforme descrito no tópico relacionado para o cliente do Java.
  - Todos os repositórios de chaves foram criados e manipulados usando somente software compatível com FIPS, como **runmqakm** com a opção `-fips`.
- Para aplicativos IBM MQ classes for JMS que usam o modo cliente, a conexão usa as implementações TLS do JRE e será compatível com o FIPS se as condições forem atendidas:
  - O Java Runtime Environment usado para executar o aplicativo for compatível com o FIPS na versão do sistema operacional e arquitetura de hardware instaladas.
  - Você especificou que somente a criptografia certificada para FIPS deve ser usada, conforme descrito no tópico relacionado para o cliente do JMS.
  - Todos os repositórios de chaves foram criados e manipulados usando somente software compatível com FIPS, como **runmqakm** com a opção `-fips`.
- Para aplicativos clientes .NET não gerenciados, a conexão usa GSKit e é compatível com FIPS se as condições a seguir forem atendidas:
  - A versão instalada do GSKit ICC foi certificada compatível com FIPS 140-2 na versão do sistema operacional e arquitetura de hardware instalados.
  - Você especificou que somente a criptografia certificada para FIPS deve ser usada, conforme descrito no tópico relacionado para o cliente do .NET.
  - Todos os repositórios de chaves foram criados e manipulados usando somente software compatível com FIPS, como **runmqakm** com a opção `-fips`.
  - O acesso a todos os repositórios de chave é fornecido usando um arquivo stash e não o mecanismo de senha do repositório de chave
- Para aplicativos clientes XMS .NET não gerenciados, a conexão usa GSKit e é compatível com FIPS se as condições a seguir forem atendidas:
  - A versão instalada do GSKit ICC foi certificada compatível com FIPS 140-2 na versão do sistema operacional e arquitetura de hardware instalados.
  - Você especificou que apenas a criptografia certificada por FIPS deve ser usada, conforme descrito na documentação do XMS .NET
  - Todos os repositórios de chaves foram criados e manipulados usando somente software compatível com FIPS, como **runmqakm** com a opção `-fips`.
  - O acesso a todos os repositórios de chave é fornecido usando um arquivo stash e não o mecanismo de senha do repositório de chave

Todas as plataformas suportadas são certificados pelo FIPS 140-2, exceto conforme observado no arquivo leia-me incluído com cada fix pack ou pacote de atualizações.

Para conexões TLS usando GSKit, o componente que é certificado pelo FIPS 140-2 é denominado *ICC*. É a versão desse componente que determina a GSKit conformidade FIPS em qualquer plataforma específica. Para determinar a versão ICC atualmente instalada, execute o comando **dspmqver -p 64 -v** .

Aqui está uma extração de exemplo da saída **dspmqver -p 64 -v** relacionada a ICC:

```
ICC
============
@ (#) CompanyName: IBM Corporation
@(#)LegalTrademarks: IBM
@ (#) FileDescription: IBM Crypto for C-language
@(#)FileVersion:   8.0.0.0
@(#)LegalCopyright:  Licensed Materials - Property of IBM
```

```
@(#)         ICC
@ (#) (C) Copyright IBM Corp. 2002, 2024.
@ (#) Todos os direitos reservados. US Government Users
@(#)         Restricted Rights - Use, duplication or disclosure
@(#)         restricted by GSA ADP Schedule Contract with IBM Corp.
@(#)ProductName:   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:
```

A instrução de certificação NIST para GSKit ICC 8 (incluída em GSKit 8) pode ser localizada no endereço a seguir: Cryptographic Module Validation Program.

Se o hardware de criptografia estiver presente, os módulos de criptografia usados pelo IBM MQ podem ser configurados para ser aqueles fornecidos pelo fabricante do hardware. Se isso for feito, a configuração só ficará em conformidade com o FIPS se esses módulos criptográficos forem certificados pelo FIPS.

Restrições do Padrão de Criptografia de Dados triplo impostas ao operar em conformidade com o FIPS 140-2

Quando o IBM MQ está configurado para operar em conformidade com o FIPS 140-2, restrições adicionais são aplicadas no que se refere ao CipherSpecs do Padrão de Criptografia de Dados triplo (3DES). Essas restrições permitem a conformidade com a recomendação do US NIST SP800-67.

1. Todas as partes do Padrão de Criptografia de Dados triplo devem ser exclusivas.
2. Nenhuma parte do Padrão de Criptografia de Dados triplo pode ser uma chave Weak, Semi-Weak ou Possibly-Weak, de acordo com as definições no NIST SP800-67.
3. Não mais que 32 GB de dados podem ser transmitidos através da conexão antes que uma reconfiguração de chave secreta deva ocorrer. Por padrão, o IBM MQ não reconfigura a chave de sessão secreta, de modo que esta redefinição deve ser configurada. Falha ao ativar a reconfiguração da chave secreta ao usar um CipherSpec do Padrão de Criptografia de Dados triplo e resultados de conformidade com o FIPS 140-2 no fechamento da conexão com o erro AMQ9288 após a contagem máxima de bytes exceder. Para obter informações sobre como definir a reconfiguração de chave secreta, consulte “Reconfigurando as chaves secretas SSL e TLS” na página 474.

O IBM MQ gera chaves de sessão do Triple DES que já cumprem as regras 1 e 2. No entanto, para atender a terceira restrição, deve-se ativar a reconfiguração de chave secreta ao usar as CipherSpecs do Triple DES em uma configuração 140-2 do FIPS. Como alternativa, é possível evitar o uso do Padrão de Criptografia de Dados triplo.

Conceitos relacionados

“Especificando que Apenas CipherSpecs Certificados por FIPS São Usados no Tempo de Execução no Cliente de MQI” na página 273
Crie seus repositórios de chaves usando software compatível com FIPS e, em seguida, especifique que o canal deve usar CipherSpecs certificados por FIPS.

Tarefas relacionadas

Ativando o TLS no IBM MQ classes for Java

Usando Segurança da Camada de Transporte (TLS) com o IBM MQ classes for JMS

Referências relacionadas

Propriedades de TLS de objetos do JMS

“Comandos runmqakm e runmqktool em AIX, Linux, and Windows” na página 550
Em sistemas AIX, Linux, and Windows , use os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool) para gerenciar chaves e certificados..

“Federal Information Processing Standards” na página 23
O governo dos EUA produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. O National Institute for Standards and Technology (NIST) é um órgão importante ligado a sistemas e segurança de TI. O NIST produz recomendações e normas, inclusive Federal Information Processing Standards (FIPS).

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

When cryptography is required on an SSL/TLS channel on z/OS , IBM MQ uses a service called System SSL. The objective of System SSL is to provide the capability to execute securely in a mode designed to adhere to the Federal Information Processing Standards (FIPS) Cryptomodule Validation Program of the US National Institute of Standards and Technology, at level 140-2.

When implementing FIPS 140-2 compliant connections with IBM MQ TLS connections there are a number of points to consider:

- To enable IBM MQ message channels for FIPS-compliance, ensure the following conditions are met:
  - System SSL Security Level 3 FMID is installed and configured (see Planning to install IBM MQ).
  - System SSL modules are validated.
  - The queue manager's SSLFIPS attribute has been set to **YES**.

When executing in FIPS mode, System SSL exploits CP Assist for Cryptographic Function (CPACF) when available. Cryptographic functions performed by ICSF-supported hardware when running in non-FIPS mode continue to be exploited when executing in FIPS mode, with the exception of RSA signature generation which must be performed in software.

Table 2. Differences between FIPS mode and non-FIPS mode algorithm support.

	Non-FIPS		**FIPS**	
Algorithm	**Key sizes**	**Hardware**	**Key sizes**	**Hardware**
RC2	40 and 128			
RC4	40 and 128			
DES	56	x		
TDES	168	x	168	x
AES	128 and 256	x	128 and 256	x
MD5	48			
SHA-1	160	x	160	x
SHA-2	224, 256, 384 and 512	x	224, 256, 384 and 512	x
RSA	512-4096	x	1024-4096	x
DSA	512-1024		1024	
DH	512-2048		2048	

In FIPS mode, System SSL can only use certificates that use the algorithms and key sizes shown in Table 1. During X.509 certificate validation if an algorithm that is incompatible with FIPS mode is encountered, then the certificate cannot be used and is treated as not valid.

For IBM MQ classes applications using client mode within WebSphere® Application Server , refer to Federal Information Processing Standard support.

For information on System SSL module configuration, see System SSL Module Verification Setup.

Related reference

“Federal Information Processing Standards” on page 23

O governo dos EUA produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. O National Institute for Standards and Technology (NIST) é um órgão importante ligado a sistemas e segurança de TI. O NIST produz recomendações e normas, inclusive Federal Information Processing Standards (FIPS).

Multi *Verificando a configuração de TLS do seu gerenciador de filas com `mqcertck`*

O comando **MQCERTCK** é uma ferramenta para procurar erros comuns na configuração de TLS do seu gerenciador de filas e fornece algumas sugestões para resolver problemas.

Introdução

O comando **`mqcertck`** verifica:

- Existência e permissões do repositório de chaves do gerenciador de filas, referenciado no atributo **SSLKEYR** do gerenciador de filas.
- Existência e validade do certificado para o certificado do gerenciador de filas, referenciado no atributo **CERTLABL** do gerenciador de filas.
- Existência e validade de quaisquer certificados referenciados nos atributos **CERTLABL** do canal ativado para TLS.
- o repositório de chaves e os certificados dos aplicativos clientes, incluindo a verificação de que os certificados estão autorizados com o gerenciador de filas.

Nota: O comando **`mqcertck`** não está disponível no z/OS ou no IBM i.

Uso

Para usar o comando **`mqcertck`**, execute o comando `mqcertck`, juntamente com os parâmetros necessários e quaisquer parâmetros opcionais requeridos em uma linha de comandos.

Veja mqcertck para obter uma descrição do comando e os parâmetros que o comando utiliza.

exemplo

Você acabou de configurar o `QM1` do gerenciador de filas para permitir conexões TLS de clientes que se conectam ao canal SVRCONN do gerenciador de filas.

Você está usando o recurso de vários certificados e, portanto, o gerenciador de filas e o canal têm um rótulo de certificado especificado em seus atributos **CERTLABL**. Ao criar o canal, você cometeu um erro no atributo **CERTLABL** do canal, portanto, quando um cliente tenta se conectar, o gerenciador de filas retorna um código de retorno 2393 de MQRC_SSL_INITIALIZATION_ERROR.

Antes de ativar o gerenciador de filas, você usa o comando **`mqcertck`** para verificar a configuração de TLS do gerenciador de filas.

Você executa o comando `mqcertck QM1` e recebe a saída a seguir:

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

```
| If there are problems then resolve these and run this
| tool again.
|
+-------------------------------------------------------
```

Essa saída solicita que você verifique a definição do seu canal para o canal de conexão do servidor MQCERTCK.CHANNEL. Aqui, você vê o erro que cometeu e pode corrigi-lo antes de executar o comando `mqcertck` novamente para verificar se resolveu o problema.

Verificando as conexões do cliente

O comando **mqcertck** tem a capacidade de verificar os repositórios de chaves do cliente, bem como a configuração de TLS do gerenciador de filas. Para fazer isso, o **mqcertck** precisa ser capaz de acessar o repositório de chaves do cliente na máquina executando o gerenciador de filas.

Ao executar o comando **mqcertck**, se você fornecer o parâmetro **-clientkeyr** com o local do repositório de chaves do cliente (exceto a extensão), o **mqcertck** verificará esse repositório de chaves com relação ao gerenciador de filas.

Se você souber qual canal o cliente usará para se conectar ao gerenciador de filas, poderá especificar isso com o sinalizador **-clientchannel**.

Se o cliente estiver usando a autenticação mútua para se conectar ao gerenciador de filas, será possível usar o parâmetro **-clientusername** ou **-clientlabel** para informar ao comando **mqcertck** qual certificado usar no repositório de chaves do cliente.

Se você estiver usando o certificado padrão e não fornecer um rótulo de certificado ao aplicativo cliente, poderá usar os parâmetros **-clientusername** e **username** que executam esse aplicativo.

Durante a operação do comando **mqcertck**, ele gera o rótulo certificado `ibmwebspheremqXXXX`, em que XXXX é o valor transmitido no parâmetro **-clientusername**.

Para verificar totalmente o repositório de chaves do cliente, o comando **mqcertck** cria uma conexão simulada usando IBM Global Security Kit (GSKit) Para fazer isso, o comando precisa ter uma porta disponível à qual possa se ligar durante os testes do cliente. A porta padrão usada é 5857, no entanto, se ela já estiver em uso, será possível especificar uma porta diferente a ser usada durante os testes do cliente.

Nota: Embora o comando **mqcertck** seja ligado a uma porta, nenhuma comunicação externa é usada pelo **mqcertck** e todos os testes são realizados localmente.

SSL/TLS no IBM MQ MQI client

O IBM MQ suporta o TLS em clientes. É possível customizar o uso do TLS de várias formas.

O IBM MQ fornece suporte TLS para o IBM MQ MQI clients em sistemas AIX, Linux, and Windows. Se você estiver usando o IBM MQ classes for Java, consulte Usando o IBM MQ classes for Java e se você estiver usando o IBM MQ classes for JMS, consulte Usando o IBM MQ classes for JMS. O restante desta seção não se aplica aos ambientes do Java ou JMS.

É possível especificar o repositório de chaves para um IBM MQ MQI client com o valor MQSSLKEYR no arquivo de configuração do cliente do IBM MQ ou quando o aplicativo faz uma chamada MQCONNX. Você tem três opções para especificar que um canal usa o TLS:

- Utilizando uma tabela de definição de canais
- Usando a estrutura de opções de configuração do SSL, MQSCO, ou uma chamada de MQCONNX
- Usando o Active Directory (nos sistemas Windows)

Não é possível usar a variável de ambiente MQSERVER para especificar que um canal usa o TLS.

É possível continuar executando seus aplicativos IBM MQ MQI client existentes sem o TLS, contanto que o TLS não seja especificado na outra extremidade do canal.

Se as mudanças forem feitas em uma máquina do cliente no conteúdo do Repositório de chaves do TLS, do local do Repositório de chaves do TLS, das Informações de autenticação ou dos parâmetros de hardware criptográficos, será necessário encerrar todas as conexões do TLS para refletir essas mudanças

nos canais de conexão do cliente que o aplicativo estiver usando para se conectar ao gerenciador de filas. Assim que todas as conexões tiverem sido encerradas, reinicie os canais do TLS. Todas as novas configurações do TLS são usadas. Essas configurações são análogas àquelas atualizadas pelo comando REFRESH SECURITY TYPE(SSL) em sistemas do Gerenciadores de Filas.

Quando o seu IBM MQ MQI client é executado um sistema AIX, Linux, and Windows com hardware criptográfico, você configura esse hardware com a variável de ambiente MQSSLCRYP. Esta variável é equivalente ao parâmetro SSLCRYP no comando ALTER QMGR MQSC. Consulte ALTER QMGR para obter uma descrição do parâmetro SSLCRYP no comando ALTER QMGR MQSC. Se você usar a versão GSK_PCS11 do parâmetro SSLCRYP, o rótulo do token PKCS #11 deverá ser especificado inteiramente em minúsculas.

A reconfiguração de chave secreta do TLS e o FIPS são suportados no IBM MQ MQI clients. Para obter mais informações, consulte “Reconfigurando as chaves secretas SSL e TLS” na página 474 e “Federal Information Processing Standards (FIPS) para AIX, Linux, and Windows” na página 36.

Consulte “Configurando a Segurança do IBM MQ MQI client” na página 272 para obter mais informações sobre o suporte do TLS para IBM MQ MQI clients.

Tarefas relacionadas

IBM MQ MQI client arquivo de configuração, `mqclient.ini`

Especificando que um canal MQI usa SSL/TLS

Para um canal MQI usar TLS, o valor do atributo *SSLCipherSpec* do canal de conexão do cliente deve ser o nome de um CipherSpec que seja suportado pelo IBM MQ na plataforma do cliente.

É possível definir um canal de conexão do cliente com um valor para este atributo das seguintes maneiras. Eles são listados na ordem de precedência decrescente.

1. Quando uma saída PreConnect fornece uma estrutura de definição de canal para uso.

   Uma saída PreConnect pode fornecer o nome de um CipherSpec no campo *SSLCipherSpec* de uma estrutura de definição de canal, MQCD. Esta estrutura é retornada no campo **ppMQCDArrayPtr** da estrutura do parâmetro de saída MQNXP usada pela saída PreConnect.
2. Quando um aplicativo do IBM MQ MQI client emite uma chamada MQCONNX.

   O aplicativo pode especificar o nome de um CipherSpec no campo *SSLCipherSpec* de uma estrutura de definição de canal, MQCD.Esta estrutura é referenciada pela estrutura de opções de conexão, MQCNO, que é um parâmetro na chamada MQCONNX.
3. Usando uma Tabela de Definição de Canal de Cliente (CCDT).

   Uma ou mais entradas em uma tabela de definição de canal do cliente podem especificar o nome de um CipherSpec.Por exemplo, se você criar uma entrada usando o comando DEFINE CHANNEL MQSC, poderá usar o parâmetro SSLCIPH no comando para especificar o nome de um CipherSpec.
4. Usando um Active Directory no Windows.

   Nos sistemas Windows, é possível usar o comando de controle **setmqscp** para publicar as definições de canal de conexão do cliente no Active Directory. Uma ou mais destas definições podem especificar o nome de um CipherSpec.

Por exemplo, se um aplicativo cliente fornece uma definição de canal de conexão do cliente em uma estrutura MQCD em uma chamada MQCONNX, esta definição será usada como preferência em qualquer entrada em uma tabela de definição de canal do cliente que pode ser acessada pelo cliente IBM MQ.

Não é possível usar a variável de ambiente MQSERVER para fornecer a definição de canal na extremidade do cliente de um canal MQI que usa TLS.

Para verificar se um certificado de cliente fluiu, exiba o status do canal na extremidade do servidor de um canal para a presença de um valor de parâmetro de nome de mesmo nível.

Conceitos relacionados

“Especificando um CipherSpec para um IBM MQ MQI client” na página 451
Você tem três opções para especificar um CipherSpec para um IBM MQ MQI client.

CipherSpecs e CipherSuites no IBM MQ

O IBM MQ suporta as CipherSpecs de TLS1.3 e TLS 1.2 e algoritmos RSA e Diffie-Hellman. No entanto, será possível ativar CipherSpecs descontinuadas, se você precisar fazer isso.

Consulte "Ativando CipherSpecs" na página 427 para obter informações sobre:

- CipherSpecs suportados pelo IBM MQ.
- Como ativar especificações de código descontinuadas do SSL 3.0 e do TLS 1.0.

O IBM MQ suporta o RSA e Diffie-Hellman Key Exchange e os algoritmos de autenticação. O tamanho da chave usada durante o handshake TLS pode depender do certificado digital usado, mas alguns CipherSpecs incluem uma especificação do tamanho de chave de handshake. Tamanhos maiores de chaves de handshake fornecem autenticação mais consistente. Com tamanhos de chaves menores, o protocolo de reconhecimento é mais veloz.

Conceitos relacionados

"CipherSpecs e CipherSuites" na página 22
Os protocolos de segurança criptográficos devem concordar sobre os algoritmos usados por uma conexão segura. CipherSpecs e CipherSuites definem combinações específicas de algoritmos.

Criptografia do Conjunto B da NSA no IBM MQ

Este tópico fornece informações sobre como configurar o IBM MQ for AIX, Linux, and Windows para se adequar ao perfil TLS 1.2 compatível com o Conjunto B.

Com o tempo, o Padrão do Conjunto B de Criptografia da NSA é atualizado para refletir novos ataques contra algoritmos de criptografia e protocolos. Por exemplo: alguns CipherSpecs podem deixar de serem certificados pelo Conjunto B. Quando tais mudanças ocorrem, o IBM MQ também é atualizado para implementar o padrão mais recente. Como um resultado, você poderá ver as mudanças no comportamento após a aplicação da manutenção. O arquivo leia-me do IBM MQ lista a versão do Conjunto B cumprido por nível de manutenção do produto. Se você configurar o IBM MQ para aplicar conformidade do Conjunto B, sempre consulte o arquivo leia-me ao planejar aplicar manutenção. Consulte IBM MQ, WebSphere MQe MQSeries leituras do produto.

Em sistemas AIX, Linux, and Windows, o IBM MQ pode ser configurado para se adequar ao perfil TLS 1.2 compatível com o Conjunto B, nos níveis de segurança mostrados na Tabela 1.

Tabela 3. Níveis de Segurança do Conjunto B com CipherSpecs Permitidos e Algoritmos de Assinatura Digital

Níveis de segurança	**CipherSpecs Permitidos**	**Algoritmos de assinatura digital**
128 bits	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA com SHA-256 ECDSA com SHA-384
192 bits	ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA com SHA-384
Ambos [1]	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA com SHA-256 ECDSA com SHA-384

1. É possível configurar ambos os níveis de segurança de 128 e 192 bits simultaneamente. Como a configuração do Conjunto B determina os algoritmos criptográficos mínimos aceitáveis, a configuração de ambos os níveis de segurança é equivalente a configurar apenas o nível de segurança de 128 bits. Os algoritmos criptográficos do nível de segurança de 192 bits são mais fortes do que o mínimo requerido para o nível de segurança de 128 bits, portanto, eles são permitidos para o nível de segurança de 128 bits, mesmo se o nível de segurança 192 bits não estiver ativado.

Nota: As convenções de nomenclatura usadas para o nível de segurança não representam necessariamente o tamanho da curva elíptica ou o tamanho da chave do algoritmo de criptografia AES.

configuração do Conjunto B para CipherSpec

Embora o comportamento padrão do IBM MQ não seja cumprir o padrão do Conjunto B, o IBM MQ pode ser configurado para se adequar a qualquer um o ambos os níveis de segurança em sistemas AIX, Linux, and Windows. Após a configuração bem-sucedida do IBM MQ para usar com o Conjunto B, qualquer tentativa de iniciar um canal de saída usando um CipherSpec que não seja compatível com o Conjunto B resulta no erro AMQ9282. Esta atividade também resulta no retorno do código de razão MQRC_CIPHER_SPEC_NOT_SUITE_B por parte do cliente do MQI. Assim como tentar iniciar um canal de entrada usando um CipherSpec que não esteja em conformidade com os resultados de configuração do Conjunto B no erro AMQ9616.

Para obter mais informações sobre o IBM MQ CipherSpecs, consulte “Ativando CipherSpecs” na página 427

Conjunto B e Certificados Digitais

O Conjunto B restringe os algoritmos de assinatura digital que podem ser usados para assinar certificados digitais. Conjunto B também restringe o tipo de chave pública que certificados pode conter. Portanto, o IBM MQ deve ser configurado para usar certificados digitais cujo algoritmo de assinatura digital e tipo de chave pública são permitidos pelo nível de segurança do Conjunto B configurado do parceiro remoto. Certificados digitais que não cumprirem os requisitos de nível de segurança serão rejeitados e a conexão falhará com o erro AMQ9633 ou AMQ9285.

Para o nível de segurança do Conjunto B de 128 bits, a chave pública do assunto do certificado é necessária para usar a curva elíptica P-256 NIST ou a curva elíptica P-384 NIST e ser assinada com a curva elíptica P-256 NIST ou a curva elíptica P-384 NIST. No nível de segurança de 192 bits do Conjunto B, a chave pública do assunto do certificado deve usar e ser assinada pela curva elíptica NIST P-384.

Para obter um certificado adequado a operações compatíveis com o Conjunto B, use o comando **runmqakm** e especifique o parâmetro **-sig_alg** para solicitar um algoritmo de assinatura digital apropriado. Os valores de parâmetro EC_ecdsa_with_SHA256 e EC_ecdsa_with_SHA384 **-sig_alg** correspondem às chaves de curva elíptica assinadas pelos algoritmos de assinatura digital permitidos peloConjunto B.

Para obter informações adicionais sobre o comando **runmqakm**, consulte “Gerenciando chaves e certificados no AIX, Linux, and Windows” na página 549.

Criando e Solicitando Certificados Digitais

Para criar um certificado digital autoassinado para testes com o Conjunto B, consulte “Criando um certificado pessoal autoassinado no AIX, Linux, and Windows” na página 551

Para solicitar um certificado digital assinado pela CA para uso com o Conjunto B, consulte “Solicitando um certificado pessoal no AIX, Linux, and Windows” na página 553.

Nota: A autoridade de certificação que está sendo usada deve gerar os certificados digitais que satisfazem os requisitos descritos em IETF RFC 6460.

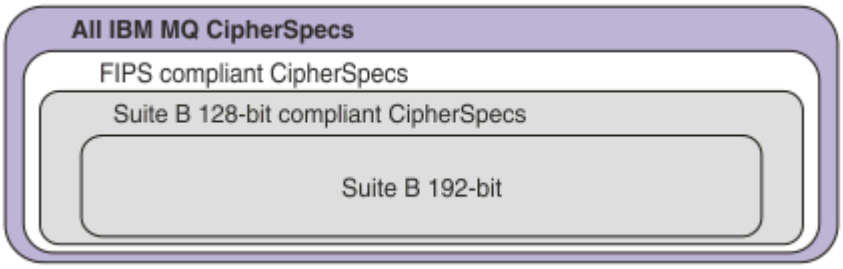
FIPS 140-2 e Conjunto B

Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o IBM Crypto for C (ICC) certificado e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na NIST CMVP modules in process list.

O IBM MQ Operator 3.2.0 e a imagem do contêiner do gerenciador de filas 9.4.0.0 em diante são baseados no UBI 9 A conformidade do FIPS 140-3 está pendente atualmente e seu status pode ser visualizado procurando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" no NIST CMVP modules in process list.

O padrão do Conjunto B é conceitualmente semelhante a FIPS 140-2, já que restringe o conjunto de algoritmos criptográficos ativados para fornecer um nível de garantia de segurança. Os CipherSpecs do Conjunto B atualmente suportados podem ser usados quando o IBM MQ é configurado para operação em conformidade com FIPS 140-2. Portanto, é possível configurar o IBM MQ para o FIPS e o Conjunto B em conformidade simultaneamente, em cujo caso os dois conjuntos de restrições se aplicam.

O diagrama a seguir ilustra a relação entre esses subconjuntos:



Configurando o IBM MQ para operação compatível com o Conjunto B.

Para obter informações sobre como configurar o IBM MQ no AIX, Linux, and Windows para operação compatível com o Conjunto B, consulte “Configurando o IBM MQ para o Conjunto B” na página 45.

O IBM MQ não suporta a operação compatível com Suite B nas plataformas e clientes a seguir:

- Plataforma do IBM i
- Plataforma do z/OS
- Java client
- JMS client

Conceitos relacionados

“Especificando que Apenas CipherSpecs Certificados por FIPS São Usados no Tempo de Execução no Cliente de MQI” na página 273
Crie seus repositórios de chaves usando software compatível com FIPS e, em seguida, especifique que o canal deve usar CipherSpecs certificados por FIPS.

ALW *Configurando o IBM MQ para o Conjunto B*

O IBM MQ pode ser configurado para operar em conformidade com o padrão do Conjunto B do NSA em plataformas AIX, Linux, and Windows.

O Conjunto B restringe o conjunto de algoritmos criptográficos ativados, a fim de fornecer um nível de segurança seguro. O IBM MQ pode ser configurado para operar em conformidade com o Conjunto B para fornecer um nível de segurança aprimorado. Para obter informações adicionais sobre o Conjunto B, consulte “Criptografia do Conjunto B da Agência Nacional de Segurança (NSA)” na página 24. Para obter mais informações sobre a configuração do Conjunto B e seu efeito sobre os canais TLS, veja “Criptografia do Conjunto B da NSA no IBM MQ” na página 43.

Gerenciador de filas

Para um gerenciador de filas, use o comando **ALTER QMGR** com o parâmetro **SUITEB** para configurar os valores apropriados para o nível de segurança requerido. Para obter mais informações, consulte ALTER QMGR.

Também é possível usar o comando **MQCMD_CHANGE_Q_MGR** de PCF com o parâmetro **MQIA_SUITE_B_STRENGTH** para configurar o gerenciador de filas para operação compatível com o Conjunto B.

Nota: Se você alterar as configurações do Conjunto B de um gerenciador de filas, deverá reiniciar o serviço MQXR para que essas configurações entrem em vigor.

Cliente MQI

Por padrão, os clientes MQI não aplicam a conformidade do Conjunto B. É possível ativar o cliente MQI para conformidade com o Conjunto B, executando uma das seguintes opções:

1. Configurando o campo EncryptionPolicySuiteB na estrutura de MQSCO em uma chamada MQCONNX para um ou mais dos seguintes valores:
   - `MQ_SUITE_B_NONE`
   - `MQ_SUITE_B_128_BIT`
   - `MQ_SUITE_B_192_BIT`

   Usar `MQ_SUITE_B_NONE` com qualquer outro valor é inválido.

   Para obter mais informações sobre a estrutura MQSCO, consulte Opções de configuração MQSCO-SSL
2. Configurando a variável de ambiente **MQSUITEB** para um ou mais dos seguintes valores:
   - `NONE`
   - `128_BIT`
   - `192_BIT`

   É possível especificar vários valores usando uma lista separada por vírgula. Usar o valor `NONE` com qualquer outro valor é inválido.
3. Configurando o atributo **EncryptionPolicySuiteB** na sub-rotina SSL do arquivo de configuração do cliente para um ou mais dos valores a seguir:
   - `NONE`
   - `128_BIT`
   - `192_BIT`

   É possível especificar vários valores usando uma lista separada por vírgula. Usar `NONE` com qualquer outro valor é inválido.

Nota: As configurações do cliente de MQI são listadas em ordem de prioridade. A estrutura MSCO na chamada MQCONNX substitui a configuração na variável de ambiente **MQSUITEB** , que substitui o atributo na sub-rotina SSL.

.NET

Para clientes não gerenciados do .NET, a propriedade **MQC.ENCRYPTION_POLICY_SUITE_B** indica o tipo de segurança do Conjunto B necessária.

Para obter informações sobre o uso do Conjunto B no IBM MQ classes for .NET, consulte Classe MQEnvironment do .NET.

AMQP

As configurações de atributo do Conjunto B para um gerenciador de filas são aplicadas a canais de AMQP nesse gerenciador de filas. Se você modificar as configurações do Conjunto B do gerenciador de filas, deverá reiniciar o serviço AMQP para que as mudanças entrem em vigor.

Políticas de validação de certificado no IBM MQ

A política de validação de certificado determina com qual precisão a validação de cadeia de certificados está em conformidade com os padrões de segurança do segmento de mercado.

A política de validação de certificado depende da plataforma e do ambiente, como a seguir:

- Para os aplicativos do Java e do JMS em todas as plataformas, a política de validação de certificado depende do componente JSSE do ambiente de tempo de execução do Java. Para obter informações adicionais sobre a política de validação de certificado, consulte a documentação do seu JRE.

- ALW Para sistemas AIX, Linux, and Windows , a política de validação de certificado é fornecida pelo IBM Global Security Kit (GSKit) e pode ser configurada.. V 9.4.0 V 9.4.0 Três diferentes políticas de validação de certificado são suportadas:
  - Uma política de validação de certificado de legado, usada para máxima compatibilidade reversa e interoperabilidade com certificados digitais antigos que não estão em conformidade com os padrões de validação de certificado IETF atuais. Esta política é conhecida como a política Básica.
  - Uma política de validação de certificado rígida e em conformidade com padrões que impinge o padrão RFC 5280. Esta política é conhecida como a política Padrão.
  - V 9.4.0 V 9.4.0 Uma política de validação de certificado que não autentica o certificado do servidor TLS, disponível apenas para aplicativos clientes
- IBM i Para sistemas IBM i, a política de validação de certificado depende da biblioteca de soquetes seguros fornecida pelo sistema operacional. Para obter informações adicionais sobre a política de validação de certificado, consulte a documentação do sistema operacional.
- z/OS Para sistemas z/OS, a política de validação de certificado depende do componente SSL do Sistema fornecido pelo sistema operacional. Para obter informações adicionais sobre a política de validação de certificado, consulte a documentação do sistema operacional.

Para obter informações sobre como configurar a política de validação de certificado, consulte “Configurando políticas de validação de certificado no IBM MQ” na página 47.. Para obter mais informações sobre as diferenças entre as políticas de validação de certificado Básicas e Padrão, consulte Validação de certificado e design de política de confiança no AIX, Linux, and Windows.

Configurando políticas de validação de certificado no IBM MQ

Há várias maneiras diferentes nas quais é possível especificar qual política de validação de certificado TLS é usada para validar certificados digitais recebidos de sistemas parceiros remotos.

Sobre esta tarefa

A política de validação de certificado determina com qual precisão a validação de cadeia de certificados está em conformidade com os padrões de segurança do segmento de mercado. A política de validação do certificado depende da plataforma e do ambiente. Para obter informações adicionais sobre as políticas de validação de certificado, consulte “Políticas de validação de certificado no IBM MQ” na página 46.

Procedimento

- Para configurar a política de validação de certificado no gerenciador de fila, use o atributo do gerenciador de fila **CERTVPOL**

  Para obter mais informações sobre como configurar esse atributo, consulte ALTER QMGR (alterar configurações do gerenciador de filas).
- Para configurar a política de validação de certificado no cliente, use os seguintes métodos.

  Se mais de um método é usado para definir a política, o cliente usará as configurações na seguinte ordem de prioridade:

  1. Use o campo `CertificateValPolicy` na estrutura MQSCO cliente. Configure o campo para um dos seguintes valores:

     MQ_CERT_VAL_POLICY_ANY
     Aplique cada uma das políticas de validação de certificado suportadas pelo secure sockets library. Aceite a cadeia de certificados se qualquer uma das políticas considerar a cadeia de certificados válida

     MQ_CERT_VAL_POLICY_RFC5280
     Aplique apenas a política de validação de certificado compatível com o RFC5280 Esta configuração fornece validação mais estrita do que a configuração ANY, mas rejeita alguns certificados digitais mais antigos.

V 9.4.0 V 9.4.0 **MQ_CERT_VAL_POLICY_NONE**
Não aplicar nenhuma política de validação de certificado Essa configuração é apenas para aplicativos clientes e aceita o certificado do servidor TLS sem validar a cadeia de confiança.

Para obter informações adicionais sobre o uso desse campo consulte MQSCO - Opções de configuração SSL.

2. Use a variável de ambiente do cliente **MQCERTVPOL**. Para configurar essa variável de ambiente, use um dos seguintes comandos:
   - Linux AIX Para sistemas AIX and Linux:

     ```
     export MQCERTVPOL= value
     ```

   - Windows Para sistemas Windows:

     ```
     SET MQCERTVPOL= value
     ```

   - IBM i Para sistemas IBM i:

     ```
     ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
     ```

3. Use o atributo **CertificateValPolicy** da sub-rotina SSL no arquivo de configuração do cliente.. Configure esse atributo para um dos seguintes valores:

   QUALQUER
   Use qualquer política de validação de certificado suportadas pela biblioteca de soquetes seguros subjacentes. Essa é a configuração padrão.

   RFC5280
   Use somente certificado de validação que está em conformidade com o padrão RFC 5280.

   V 9.4.0 V 9.4.0 **Nenhum**
   Não aplicar nenhuma política de validação de certificado Essa configuração aceita o certificado do servidor TLS sem validar a cadeia de confiança

   Para obter mais informações sobre como usar esse atributo, consulte Sub-rotina SSL do arquivo de configuração do cliente

Certificados digitais e compatibilidade de CipherSpec no IBM MQ

Este tópico fornece informações sobre como escolher os certificados digitais e do CipherSpecs para sua política de segurança, delineando o relacionamento entre os certificados digitais e de CipherSpecs no IBM MQ.

Somente um subconjunto dos CipherSpecs suportados pode ser usado com todos os tipos suportados de certificado digital. Portanto, é necessário escolher um CipherSpec apropriado para o seu certificado digital. Da mesma forma, se a política de segurança de sua organização requer o uso de um CipherSpec específico, deve-se obter um certificado digital apropriado para esse CipherSpec.

Os algoritmos de assinatura digital MD5 e o TLS 1.2

Os certificados digitais assinados usando o algoritmo MD5 são rejeitados quando o protocolo TLS 1.2 é usado. Isso é porque o algoritmo MD5 é agora considerado fraco por muitos analistas de criptografia, e seu uso é normalmente desencorajado. Para usar as CipherSpecs mais recentes com base no protocolo TLS 1.2, assegure-se de que os certificados digitais não usem o algoritmo MD5 em suas assinaturas digitais. Os CipherSpecs mais antigos que usam os protocolos TLS 1.0 não estão sujeitos a essa restrição e podem continuar a usar certificados com assinaturas digitais MD5.

Para visualizar o algoritmo de assinatura digital para um certificado específico, é possível usar o comando **runmqakm**:

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

em que *cert_label* é o rótulo do certificado do algoritmo de assinatura digital a ser exibido. Consulte Rótulos de certificado digital para obter detalhes.

A execução do comando **runmqakm** produz a saída exibindo o uso do algoritmo de assinatura especificado:

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

A linha `Signature Algorithm` mostra que o algoritmo `MD5WithRSASignature` é usado.. Este algoritmo é baseado em MD5 e, portanto, este certificado digital não pode ser usado com o CipherSpecs do TLS 1.2.

Interoperabilidade da curva elíptica e do RSA de CipherSpecs

Nem todos os CipherSpecs podem ser usados com todos os certificados digitais. CipherSpecs são denotados pelo prefixo do nome CipherSpec Cada tipo de CipherSpec impõe restrições diferentes sobre o tipo de certificado digital que pode ser utilizado Essas restrições são aplicadas a todas as conexões TLS do IBM MQ, mas são particularmente relevantes para os usuários da criptografia de Curva Elíptica.

A tabela a seguir resume os relacionamentos entre os certificados digital e de CipherSpecs:

Tabela 4. Relacionamentos entre os certificados digital e de CipherSpecs

tipo	Prefixo de nome do CipherSpec	Descrição	Tipo de chave pública requerido	Algoritmo de criptografia de assinatura digital	Método de estabelecimento de chave secreta
1	ECDHE_ECDSA_	Os CipherSpecs que usam as chaves públicas da Curva Elíptica, as chaves secretas da Curva Elíptica e os algoritmos de assinatura digital da Curva Elíptica.	Curva Elíptica	ECDSA	ECDHE
2	ECDHE_RSA_	CipherSpecs que usam chaves públicas RSA, chaves secretas de Curva Elíptica e algoritmos de assinatura digital RSA.	RSA	RSA	ECDHE
3	(Todos os TLS 1.3 CipherSpecs)	CipherSpecs que usam chaves públicas Elliptic Curve ou RSA, chaves secretas Elliptic Curve e algoritmos de assinatura digital Elliptic Curve ou RSA.	Curva elíptica ou RSA	ECDSA ou RSA	ECDHE ou RSA
4	(Todos os outros)	Os CipherSpecs que usam chaves públicas de RSA e algoritmos de assinatura digital de RSA.	RSA	RSA	RSA

Nota: Os CipherSpecs do tipo 1 e 2 não são suportados pelos gerenciadores de filas do IBM MQ e clientes do MQI na plataforma do IBM i.

A coluna de tipo de chave pública necessária mostra o tipo de chave pública que o certificado pessoal deve ter ao usar cada tipo de CipherSpec. O certificado pessoal é o certificado de entidade final que identifica o gerenciador de filas ou cliente para seu parceiro remoto.

Deve-se assegurar que o certificado que é nomeado no rótulo certificado seja apropriado para o canal de CipherSpec. Ou seja, se você configurar um canal com um CipherSpec que requeira um certificado de Curva Elíptica (EC), não será possível nomear um certificado RSA no rótulo certificado. Se você configurar um canal com um CipherSpec que requeira um certificado RSA, não será possível nomear um certificado EC no rótulo certificado.

Supondo que você tenha configurado corretamente o IBM MQ, será possível ter:

- Um gerenciador de filas único com uma mistura de certificados RSA e EC.
- Canais diferentes no mesmo gerenciador de filas usando um certificado RSA ou EC.

O algoritmo de criptografia de assinatura digital se refere ao algoritmo de criptografia usado para validar o peer. O algoritmo de criptografia é usado juntamente com um algoritmo hash, como MD5, SHA-1 ou SHA-256 para calcular a assinatura digital. Há vários algoritmos de assinatura digital que podem ser usados, por exemplo, RSA com MD5 ou ECDSA com SHA-256. Na tabela, ECDSA refere-se ao conjunto de algoritmos de assinatura digital que usam ECDSA; RSA refere-se ao conjunto de algoritmos de assinatura digital que usam RSA. Qualquer algoritmo de assinatura digital suportado no conjunto pode ser usado, contanto que seja baseado no algoritmo de criptografia indicado.

Os CipherSpecs do tipo 1 requerem que o certificado pessoal tenha uma chave pública da Curva Elíptica. Quando estes CipherSpecs são usados, o acordo da chave efêmera de Diffie Hellman de Curva Elíptica é usado para estabelecer a chave secreta para a conexão.

Os CipherSpecs do tipo 2 requerem que o certificado pessoal tenha uma chave pública de RSA. Quando estes CipherSpecs são usados, o acordo da chave efêmera de Diffie Hellman de Curva Elíptica é usado para estabelecer a chave secreta para a conexão.

Os CipherSpecs de tipo 3 requerem que o certificado pessoal tenha uma chave pública de RSA. Quando estes CipherSpecs são usados, a troca de chave de RSA é usada para estabelecer a chave secreta para a conexão.

Esta lista de restrições não é completa: dependendo da configuração, pode haver restrições adicionais que podem afetar ainda mais a capacidade de interoperar. Por exemplo, se IBM MQ for configurado para estar em conformidade com os padrões FIPS 140-2 ou Conjunto B da NSA, isto também limitará o intervalo de configurações permitidas. Consulte a seção seguinte para obter informações adicionais.

Se precisar usar diferentes tipos de CipherSpec no mesmo gerenciador de filas ou aplicativo cliente, configure um rótulo do certificado apropriado e a combinação do CipherSpec na definição do cliente.

Os três tipos de CipherSpec não interoperam diretamente: esta é uma limitação dos padrões de TLS atuais. Por exemplo, suponha que você tenha escolhido usar o ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec para um canal receptor denominado TO.QM1 em um gerenciador de filas denominado QM1, então o receptor deve ter um certificado pessoal com uma chave de Curva Elíptica e uma assinatura digital baseada em ECDSA. Se o canal receptor não atender a esses requisitos, o canal falhará ao iniciar.

Outros canais que se conectam com o gerenciador de filas QM1 podem usar outros CipherSpecs, desde que cada canal use um certificado do tipo correto para o CipherSpec desse canal. Por exemplo, suponha que QM1 use um canal emissor denominado TO.QM2 para enviar mensagens para outro gerenciador de filas denominado QM2. O canal TO.QM2 pode usar o CipherSpec do tipo 3 TLS_RSA_WITH_AES_256_CBC_SHA256, desde que ambas as extremidades do canal usem certificados que contêm chaves públicas de RSA. O atributo do canal do rótulo de certificado pode ser usado para configurar um certificado diferente para cada canal.

Ao planejar as redes de seu IBM MQ, considere cuidadosamente quais canais requerem TLS e certifique-se de que o tipo de certificados usados para cada canal sejam apropriados para uso com o CipherSpec naquele canal.

Para visualizar o algoritmo de assinatura digital e o tipo de chave pública de um certificado digital, é possível usar o comando **runmqakm**:

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

em que *cert_label* é o rótulo do certificado cujo algoritmo de assinatura digital precisa ser exibido. Consulte Rótulos de certificado digital para obter detalhes.

A execução do comando **runmqakm** produzirá a saída exibindo o Tipo de Chave Pública:

```
Label : ibmmqexample
Key Size : 384
Version : X509 V3
Serial : 9ad5eeef5d756f41
Issuer : CN=Example Certificate Authority,OU=Test,O=Example,C=US
Subject : CN=Example Queue Manager,OU=Test,O=Example,C=US
Not Before : 21 August 2011 13:10:24 GMT+01:00
Not After : 21 August 2012 13:10:24 GMT+01:00
Public Key
    30 76 30 10 06 07 2A 86 48 CE 3D 02 01 06 05 2B
    81 04 00 22 03 62 00 04 3E 6F A9 06 B6 C3 A0 11
    F8 D6 22 78 FE EF 0A FE 34 52 C0 8E AB 5E 81 73
    D0 97 3B AB D6 80 08 E7 31 E9 18 3F 6B DE 06 A7
    15 D6 9D 5B 6F 56 3B 7F 72 BB 6F 1E C9 45 1C 46
    60 BE F2 DC 1B AD AC EC 64 4C 0E 06 65 6E ED 93
    B8 F5 95 E0 F9 2A 05 D6 21 02 BD FB 06 63 A1 CC
    66 C6 8A 0A 5C 3F F7 D3
Public Key Type : EC_ecPublicKey (1.2.840.10045.2.1)
Fingerprint : SHA1 :
    3C 34 58 04 5B 63 5F 5C C9 7A E7 67 08 2B 84 43
    3D 43 7A 79
Fingerprint : MD5 :
    49 13 13 E1 B2 AC 18 9A 31 41 DC 8C B4 D6 06 68
Fingerprint : SHA256 :
```

```
    6F 76 78 68 F3 70 F1 53 CE 39 31 D9 05 C5 C5 9F
    F2 B8 EE 21 49 16 1D 90 64 6D AC EB 0C A7 74 17
Signature Algorithm : EC_ecdsa_with_SHA384 (1.2.840.10045.4.3.3)
Value
    30 65 02 30 0A B0 2F 72 39 9E 24 5A 22 FE AC 95
    0D 0C 6D 6C 2F B3 E7 81 F6 C1 36 1B 9A B0 6F 07
    59 2A A1 4C 02 13 7E DD 06 D6 FE 4B E4 03 BC B1
    AC 49 54 1E 02 31 00 90 0E 46 2B 04 37 EE 2C 5F
    1B 9C 69 E5 99 60 84 84 10 71 1A DA 63 88 33 E2
    22 CC E6 1A 4E F4 61 CC 51 F9 EE A0 8E F4 DC B5
    0B B9 72 58 C3 C7 A4
Trust Status : Enabled
```

A linha Tipo de Chave Pública neste caso mostra que o certificado tem uma chave pública de Curva Elíptica. A linha de Algoritmo de Assinatura neste caso mostra que o algoritmo EC_ecdsa_with_SHA384 está em uso: isso é baseado no algoritmo de ECDSA. Esse certificado é, portanto, adequado apenas para uso com o CipherSpecs tipo 1.

TLS 1.3 CipherSpecs

TLS 1.3 CipherSpecs suportam certificados ECDSA e RSA.

Curva Elíptica de CipherSpecs e Conjunto B da NSA

Quando o IBM MQ é configurado para conformidade com o Conjunto B compatível com o perfil do TLS 1.2, os algoritmos de assinatura digital e de CipherSpecs permitidos são restringidos conforme descrito em “Criptografia do Conjunto B da NSA no IBM MQ” na página 43. Além disso, o intervalo de chaves aceitável do Elliptic Curve é reduzido de acordo com os níveis de segurança configurados.

No nível de segurança do Conjunto B de 128 bits, a chave pública do assunto do certificado é necessária para usar a curva elíptica de NIST P-256 ou NIST P-384 e ser assinada com a curva elíptica NIST P-256 ou NIST P-384. O comando **runmqakm** pode ser usado para solicitar certificados digitais para esse nível de segurança, usando um parâmetro -sig_alg de EC_ecdsa_with_SHA256 ou EC_ecdsa_with_SHA384.

No nível de segurança do Conjunto B de 192 bits, a chave pública do assunto do certificado é necessária para usar a curva elíptica de NIST P-384 e ser assinada com a curva elíptica NIST P-384. O comando **runmqakm** pode ser usado para solicitar certificados digitais para esse nível de segurança usando um parâmetro -sig_alg de EC_ecdsa_with_SHA384.

As curva elípticas de NIST suportadas são as seguintes:

Tabela 5. Curvas elípticas de NIST suportadas

Nome da curva NIST FIPS 186-3	**Nome da curva RFC 4492**	**Tamanho da chave da Curva Elíptica (bits)**
P-256	secp256r1	256
P-384	secp384r1	384
P-521	secp521r1	521

Nota: A curva elíptica NIST P-521 não pode ser usado para operação compatível com o Conjunto B.

Conceitos relacionados

“Ativando CipherSpecs” na página 427
Ative um CipherSpec usando o parâmetro **SSLCIPH** no comando do MQSC **DEFINE CHANNEL** ou **ALTER CHANNEL**.

“Especificando que Apenas CipherSpecs Certificados por FIPS São Usados no Tempo de Execução no Cliente de MQI” na página 273
Crie seus repositórios de chaves usando software compatível com FIPS e, em seguida, especifique que o canal deve usar CipherSpecs certificados por FIPS.

“Criptografia do Conjunto B da NSA no IBM MQ” na página 43

Este tópico fornece informações sobre como configurar o IBM MQ for AIX, Linux, and Windows para se adequar ao perfil TLS 1.2 compatível com o Conjunto B.

"Criptografia do Conjunto B da Agência Nacional de Segurança (NSA)" na página 24
O governo dos Estados Unidos da América produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. A Agência Nacional de Segurança dos Estados Unidos (NSA) recomenda um conjunto de algoritmos criptográficos interoperável em seu padrão do Conjunto B.

Registros de Autenticação de Canal

Para exercer um controle mais preciso sobre o acesso concedido à conexão de sistemas em um nível de canal, é possível usar registros de autenticação de canal.

Você pode achar que os clientes tentam se conectar ao seu gerenciador de filas usando um ID do usuário em branco ou um ID do usuário de alto nível que permitiria ao cliente executar ações indesejáveis. É possível bloquear o acesso a esses clientes usando os registros de autenticação de canal. Alternativamente, um cliente pode declarar um ID do usuário que seja válido na plataforma do cliente, mas é desconhecido ou de um formato inválido na plataforma do servidor. É possível usar um registro de autenticação de canal para mapear o ID do usuário declarado para um ID do usuário válido.

Você pode achar um aplicativo cliente que se conecta ao seu gerenciador de filas e se comporta indevidamente de algum modo. Para proteger o servidor contra os problemas que este aplicativo está causando, ele precisar ser bloqueado temporariamente usando o endereço IP no qual o aplicativo cliente está até o momento em que as regras de firewall são atualizadas ou o aplicativo cliente é corrigido. É possível usar um registro de autenticação de canal para bloquear o endereço IP a partir do qual o aplicativo cliente se conecta.

Se tiver configurado uma ferramenta de administração, tal como IBM MQ Explorer, e um canal para esse uso específico, você pode desejar assegurar que apenas computadores clientes específicos possam usá-lo. É possível usar um registro de autenticação de canal para permitir que o canal seja usado apenas a partir de determinados endereços IP.

Se você estiver apenas iniciando alguns aplicativos de amostra executando como clientes, consulte Preparando e executando os programas de amostra para obter um exemplo de como configurar o gerenciador de filas de maneira segura usando registros de autenticação de canal.

Para obter registros de autenticação de canal para controlar canais de entrada, use o comando MQSC **`ALTER QMGR CHLAUTH(ENABLED)`**.

As regras de **CHLAUTH** são aplicadas para um canal MCA que é criado em resposta a uma nova conexão de entrada. Para um canal MCA criado em resposta ao canal sendo iniciado localmente, nenhuma regra de **CHLAUTH** é aplicada.

Tabela 6. Quando as regras de CHLAUTH são aplicadas para pares de canais diferentes

Tipo de canal	**MCA no qual as regras CHLAUTH são aplicadas**
SDR-RCVR	RCVR
RQSTR-SVR (iniciado em SVR)	RQSTR
RQSTR-SVR (iniciado em RQSTR)	SVR
RQSTR-SDR (iniciado em SDR)	RQSTR
RQSTR-SDR (iniciado em RQSTR)	SDR para conexão inicial. RQSTR para conexão de retorno de chamada.

Os registros de autenticação de canal podem ser criados para executar as seguintes funções:

- Bloquear as conexões dos endereços IP específicos.
- Bloquear as conexões de IDs de usuário específicos.
- Configurar um valor MCAUSER a ser usado para qualquer canal que se conecta a partir de um endereço IP específico.

- Configurar um valor MCAUSER a ser usado para qualquer canal declarando um ID do usuário específico.
- Configurar um valor MCAUSER a ser usado para qualquer canal que tenha um SSL específico ou Nome Distinto TLS (DN).
- Configurar um valor MCAUSER a ser usado para qualquer canal que se conecta a partir de um gerenciador de filas específico.
- Bloquear as conexões consideradas de um certo gerenciador de filas, a menos que a conexão seja de um endereço IP específico.
- Bloquear conexões que apresentam um certo certificado SSL ou TLS, a menos que a conexão seja de um endereço IP específico.

Estas utilizações são explicadas em detalhes nas seções a seguir.

Você cria, modifica ou remove registros de autenticação de canais usando o comando MQSC **SET CHLAUTH** ou o comando PCF **Set Channel Authentication Record**.

Nota: Grandes números de registros de autenticação de canal podem ter um impacto negativo no desempenho de um gerenciador de filas.

Bloqueando endereços IP

Normalmente é a função de um firewall evitar o acesso de determinados endereços IP. No entanto, pode haver ocasiões nas quais você sofre tentativas de conexão de um endereço IP que não deve ter acesso ao seu sistema IBM MQ e deve bloquear temporariamente o endereço antes da atualização do firewall. Estas tentativas de conexão podem não estar vindo de canais do IBM MQ; essas tentativas de conexão podem vir de outros aplicativos de soquete que estão mal configurados para destinar seu listener do IBM MQ. Bloqueie os endereços IP configurando um registro de autenticação de canal do tipo BLOCKADDR. É possível especificar um ou mais endereços únicos, intervalos de endereços ou padrões, incluindo curingas.

Sempre que uma conexão de entrada é recusada porque o endereço IP está bloqueado desta maneira, uma mensagem do evento MQRC_CHANNEL_BLOCKED com o qualificador de razão MQRQ_CHANNEL_BLOCKED_ADDRESS é emitida, desde que os eventos do canal estejam ativados e o gerenciador de filas esteja em execução. Além disso, a conexão é mantida aberta por 30 segundos antes de retornar o erro para assegurar que o listener não estoure com tentativas repetidas de conexão que estão bloqueadas.

Para bloquear endereços IP somente em canais específicos ou para evitar o atraso antes do erro ser relatado, configure um registro de autenticação de canal do tipo ADDRESSMAP com o parâmetro USERSRC(NOACCESS).

Sempre que uma conexão de entrada é recusada por esta razão, uma mensagem do evento MQRC_CHANNEL_BLOCKED com o qualificador de razão MQRQ_CHANNEL_BLOCKED_NOACCESS é emitida, desde que os eventos do canal estejam ativados e o gerenciador de filas esteja em execução.

Consulte “Bloqueando Endereços IP Específicos” na página 390 para obter um exemplo.

Bloqueando IDs de usuário

Para evitar que determinados IDs do usuário se conectem por meio de um canal do cliente, configure o registro de autenticação de canal do tipo BLOCKUSER. Este tipo de registro de autenticação de canal se aplica somente a canais do cliente, não a canais de mensagens. É possível especificar um ou mais IDs de usuários individuais a serem bloqueados, mas você não pode usar curingas.

Sempre que uma conexão de entrada é recusada por esta razão, uma mensagem do evento MQRC_CHANNEL_BLOCKED com o qualificador de razão MQRQ_CHANNEL_BLOCKED_USERID é emitida, desde que os eventos do canal estejam ativados.

Consulte “Bloqueando IDs de Usuários Específicos” na página 392 para obter um exemplo.

Também é possível bloco qualquer acesso para IDs de usuários especificados em determinados canais configurando um registro de autenticação de canal do tipo USERMAP com o parâmetro USERSRC(NOACCESS).

Sempre que uma conexão de entrada é recusada por esta razão, uma mensagem do evento MQRC_CHANNEL_BLOCKED com o qualificador de razão MQRQ_CHANNEL_BLOCKED_NOACCESS é emitida, desde que os eventos do canal estejam ativados e o gerenciador de filas esteja em execução.

Consulte "Bloqueando o acesso para um ID de usuário cliente" na página 395 para obter um exemplo.

Bloqueando nomes do gerenciador de filas

Para especificar que qualquer canal que se conecta a partir de um gerenciador de filas especificado não deve ter acesso, configure um registro de autenticação de canal do tipo QMGRMAP com o parâmetro USERSRC(NOACCESS). É possível especificar um único nome do gerenciador de filas ou um padrão incluindo curingas. Não há equivalente da função BLOCKUSER para bloquear o acesso de gerenciadores de filas.

Sempre que uma conexão de entrada é recusada por esta razão, uma mensagem do evento MQRC_CHANNEL_BLOCKED com o qualificador de razão MQRQ_CHANNEL_BLOCKED_NOACCESS é emitida, desde que os eventos do canal estejam ativados e o gerenciador de filas esteja em execução.

Consulte "Bloqueando Acesso de um Gerenciador de Filas Remotas" na página 394 para obter um exemplo.

Bloqueando DNs de SSL ou TLS

Para especificar que qualquer usuário que apresenta um certificado pessoal de SSL ou TLS contendo um DN especificado não possui acesso, configure um registro de autenticação de canal do tipo SSLPEERMAP com o parâmetro USERSRC(NOACCESS). É possível especificar um único nome distinto ou um padrão incluindo curingas. Não há equivalente da função BLOCKUSER para bloquear o acesso para DNs.

Sempre que uma conexão de entrada é recusada por esta razão, uma mensagem do evento MQRC_CHANNEL_BLOCKED com o qualificador de razão MQRQ_CHANNEL_BLOCKED_NOACCESS é emitida, desde que os eventos do canal estejam ativados e o gerenciador de filas esteja em execução.

Consulte "Bloqueando o acesso para um Nome Distinto SSL ou TLS" na página 395 para obter um exemplo.

Mapeando endereços IP para IDs do usuário para serem usados

Para especificar se algum canal conectando-se a partir de um endereço IP especificado deve usar um MCAUSER específico, configure um registro de autenticação de canal do tipo ADDRESSMAP. É possível especificar um único endereço, um intervalo de endereços ou um padrão incluindo curingas.

Se você usar um encaminhador de porta, quebra da sessão DMZ ou qualquer outra configuração que mude o endereço IP apresentado ao gerenciador de filas, o mapeamento de endereços IP não será necessariamente adequado para uso.

Consulte "Mapeando um Endereço IP para um ID do Usuário MCAUSER" na página 396 para obter um exemplo.

Mapeando nomes do gerenciador de filas para IDs do usuário para serem usados

Para especificar se algum canal conectando-se a partir de um gerenciador de filas especificado deve usar um MCAUSER específico, configure um registro de autenticação de canal do tipo QMGRMAP. É possível especificar um único nome do gerenciador de filas ou um padrão incluindo curingas.

Consulte "Mapeando um Gerenciador de Filas Remoto para um ID do Usuário MCAUSER" na página 392 para obter um exemplo.

Mapeando IDs de Usuários Declarados por um Cliente para IDs de Usuários a Serem Usados

Para especificar que um determinado ID do usuário é usado por uma conexão a partir de um cliente IBM MQ MQI, um MCAUSER diferente especificado deve ser usado. configurar um registro de autenticação de canal do tipo USERMAP. O mapeamento do ID do usuário não usa curingas.

Consulte “Mapeando um ID de usuário cliente para um ID do usuário MCAUSER” na página 393 para obter um exemplo.

Mapeando DNs SSL ou TLS para IDs do usuário para serem usados

Para especificar se algum usuário apresentando um certificado pessoal de Secure Sockets Layer/ Segurança da Camada de Transporte contendo um Nome distinto especificado deve usar um MCAUSER específico, configure um registro de autenticação de canal do tipo SSLPEERMAP. É possível especificar um único nome distinto ou um padrão incluindo curingas.

Consulte “Mapeando um Nome Distinto de SSL ou TLS para um ID do Usuário MCAUSER” na página 394 para obter um exemplo.

Mapeamento de Gerenciadores de Filas, Clientes ou DNs de SSL ou TLS de acordo com Endereço IP

Em algumas circunstâncias, pode ser possível para um terceiro imitar um nome do gerenciador de filas. Um certificado SSL ou TLS ou arquivo do banco de dados de chave também pode ser deturpado e reutilizado. Para se proteger contra essas ameaças, é possível especificar que uma conexão a partir de um determinado gerenciador de filas ou cliente, ou usando um determinado Nome distinto deve ser estabelecida a partir de um endereço IP especificado. Configure um registro de autenticação de canal do tipo USERMAP, QMGRMAP ou SSLPEERMAP e especifique o endereço IP permitido, ou padrão de endereços IP, usando o parâmetro ADDRESS.

Consulte “Mapeando um Gerenciador de Filas Remoto para um ID do Usuário MCAUSER” na página 392 para obter um exemplo.

Interação entre registros de autenticação de canal

É possível que um canal ao tentar fazer uma conexão corresponda a mais de um registro de autenticação de canal e que isso tenha efeitos contraditórios. Por exemplo, um canal pode declarar um ID do usuário que foi bloqueado por um registro de autenticação de canal BLOCKUSER, mas com um certificado SSL ou TLS que corresponde a um registro SSLPEERMAP que configura um ID do usuário diferente. Além disso, se registros de autenticação de canal usarem curingas, um único endereço IP, nome do gerenciador de filas ou Nome distinto Secure Sockets Layer ou de Segurança da Camada de Transporte pode corresponder a vários padrões. Por exemplo, o endereço IP 192.0.2.6 corresponde aos padrões 192.0.2.0-24, 192.0.2. *, e 192.0.*.6. A ação executada é determinada conforme a seguir.

- O registro de autenticação de canal usado é selecionado conforme a seguir:
  - Um registro de autenticação de canal que corresponde explicitamente ao nome de canal tem prioridade sobre um registro de autenticação de canal que corresponde ao nome de canal usando um curinga.
  - Um registro de autenticação de canal usando um DN SSL ou TLS tem prioridade sobre um registro que usa um ID do usuário, nome do gerenciador de filas ou endereço IP.
  - Um registro de autenticação de canal que usa um ID do usuário ou um nome do gerenciador de filas tem prioridade sobre um registro que usa um endereço IP.
- Se um registro de autenticação de canal correspondente for localizado e ele especificar um MCAUSER, este MCAUSER será designado ao canal.
- Se um registro de autenticação de canal correspondente for localizado e ele especificar que o canal não possui acesso, um valor de MCAUSER igual a *NOACCESS será designado ao canal. Este valor pode, posteriormente, ser alterado por um programa de saída de segurança.
- Se nenhum registro de autenticação de canal correspondente for localizado, ou um registro de autenticação de canal correspondente for localizado e ele especificar que o ID do usuário do canal deve ser usado, o campo MCAUSER será inspecionado.
  - Se o campo MCAUSER estiver em branco, o ID do usuário do cliente é designado ao canal.
  - Se o campo MCAUSER não estiver em branco, ele será designado ao canal.

- Qualquer programa de saída de segurança é executado. Este programa de saída pode configurar o ID do usuário do canal ou determinar que o acesso deve ser bloqueado.
- Se a conexão estiver bloqueada ou o MCAUSER estiver configurado para *NOACCESS, o canal é encerrado.
- Se a conexão não estiver bloqueada, para qualquer canal exceto um canal do cliente, o ID do usuário do canal determinado nas etapas anteriores será verificado com relação à lista de usuários bloqueados.
  - Se o ID do usuário estiver na lista de usuários bloqueados, o canal será encerrado.
  - Se o ID do usuário não estiver na lista de usuários bloqueados, o canal é executado.

A correspondência mais específica é usada quando um número de registros de autenticação de canal corresponde a um nome de canal, endereço IP, nome do host, nome do gerenciador de filas ou DN de SSL ou TLS. A correspondência considerada como sendo:

- A mais específica é um nome sem caracteres curinga, por exemplo:
  - Um nome de canal de A.B.C
  - Um endereço IP de 192.0.2.6
  - Um nome de host de `hursley.ibm.com`
  - Um nome do gerenciador de filas de 192.0.2.6
- O mais genérico é um asterisco (*) único que corresponde, por exemplo:
  - Todos os nomes de canais
  - Todos os endereços IP
  - Todos os nomes de hosts
  - Todos os nomes do gerenciador de filas
- Um padrão com um asterisco no início de uma sequência é mais genérico do que um valor definido no início de uma sequência:
  - Para canais, *.B.C é mais genérico que A.*
  - Para endereços IP, *.0.2.6 é mais genérico do que 192.*
  - Para nomes de host, `*.ibm.com` é mais genérico do que `hursley.*`
  - Para nomes de gerenciadores de filas, *QUEUEMANAGER é mais genérico do que QUEUEMANAGER*
- Um padrão com um asterisco em um local específico em uma sequência é mais genérico do que um valor definido no mesmo local em uma sequência, e de forma semelhante para cada local subsequente em uma sequência:
  - Para canais, A.*.C é mais genérico do que A.B.*
  - Para endereços IP, 192.*.2.6 é mais genérico do que 192.0.*.
  - Para nomes de host, `hursley.*.com` é mais genérico do que `hursley.ibm.*`
  - Para nomes de gerenciadores de filas, Q*MANAGER é mais genérico do que QUEUE*
- Onde dois ou mais padrões possuem um asterisco em um local específico em uma sequência, aquele com menos nós após o asterisco é mais genérico:
  - Para canais, A.* é mais genérico do que A.*.C
  - Para endereços IP, 192.* é mais genérico do que 192.*.2.*.
  - Para nomes de host, `hurlsey.*` é mais genérico do que `hursley.*.com`
  - Para nomes de gerenciadores de filas, Q* é mais genérico do que Q*MGR
- Além disso, para um endereço IP:
  - Um intervalo indicado com um hífen (-) é mais específico do que com um asterisco. Portanto, 192.0.2.0-24 é mais específico do que 192.0.2.*.
  - Um intervalo que é um subconjunto de um outro é mais específico do que o intervalo maior. Portanto, 192.0.2.5-15 é mais específico do que 192.0.2.0-24.

- A sobreposição de intervalos não é permitida. Por exemplo, não é possível ter registros de autenticação de canal para 192.0.2.0-15 e 192.0.2.10-20.
- Um padrão não pode ter menos do que o número necessário de partes, a menos que o padrão termine com um único asterisco final. Por exemplo 192.0.2 é inválido, mas 192.0.2 .* é válido
- Um asterisco final deve ser separado do restante do endereço pelo separador de parte apropriado (um ponto (.) para IPv4, dois pontos (:) para IPv6). Por exemplo, 192.0* não é válido porque o asterisco não está em uma parte própria sua.
- Um padrão pode conter asteriscos adicionais, contanto que nenhum asterisco seja adjacente ao asterisco final. Por exemplo, 192.*.2.* é válido, mas 192.0.*.* não é válido.
- Um padrão de endereço IPv6 não pode conter dois pontos duplos e um asterisco final, pois o endereço resultante seria ambíguo. Por exemplo, 2001::* poderia expandir para 2001:0000:*, 2001:0000:0000:* e assim por diante.

- Para um Nome Distinto (DN) SSL ou TLS, a ordem de precedência das subsequências é a seguinte:

Tabela 7. Ordem de precedência de subsequências

Ordem	Subsequência do DN	Nome
1	SERIALNUMBER=	Número de série do certificado
2	MAIL=	Endereço de e-mail
3	Deprecated E=	Endereço de e-mail (descontinuado na preferência para MAIL)
4	UID=, USERID=	Identificador de usuários
5	CN=	Nome comum
6	T =	Título
7	OU=	unidade organizacional
8	DC=	Componente de domínio
9	O=	Organização
10	STREET=	Rua / Primeira linha do endereço
11	L=	Localidade
12	ST=, SP=, S=	Nome do estado ou território
13	PC=	Código Postal / Código de Endereçamento Postal
14	C=	País
15	UNSTRUCTUREDNAME=	Nome do host
16	UNSTRUCTUREDADDRESS=	endereço IP
17	DNQ=	Qualificador de Nome Distinto

Portanto, se um certificado SSL ou TLS for apresentado com um DN contendo as subsequências O=IBM e C=UK, o IBM MQ usará um registro de autenticação de canal para O=IBM em preferência a um para C=UK, se ambos estiverem presentes.

Um Nome distinto pode conter diversas OUs, que devem ser especificadas em ordem hierárquica com as maiores unidades organizacionais especificadas primeiro. Se dois Nomes distintos forem iguais em todos os aspectos, exceto por seus valores de OU, o Nome distinto mais específico será determinado conforme a seguir:

1. Se eles possuírem números diferentes de atributos de OU, o Nome distinto com a maioria dos valores de OU é mais específico. Isso porque o Nome distinto com mais Unidades Organizacionais qualifica integralmente o Nome distinto em mais detalhes e fornece mais critérios de correspondência. Mesmo se sua OU de nível superior for um curinga (OU=*), o DN com mais OUs ainda será considerado o mais específico no geral.
2. Se eles tiverem o mesmo número de atributos de OU, os pares correspondentes de valores de OU são comparados na sequência da esquerda-para-direita, em que a OU mais à esquerda é o nível mais superior (menos específico), de acordo com as seguintes regras.
   a. Uma OU sem nenhum valor curinga é a mais específica porque ela pode corresponder exatamente com uma sequência apenas.
   b. Uma OU com um único curinga no início ou no final (por exemplo, OU=ABC* ou OU=*ABC) é a próxima mais específica.
   c. Uma OU com dois curingas (por exemplo, OU=*ABC*) é a próxima mais específica.
   d. Uma OU que consiste em somente um asterisco (OU=*) é a menos específica.
3. Se a comparação de sequência tiver uma ligação entre dois valores de atributo com a mesma especificidade, a sequência de atributos mais longa será mais específica.
4. Se a comparação de sequência estiver empatada entre dois valores de atributo de mesma especificidade e comprimento, então o resultado será determinado por uma comparação de sequência sem distinção entre maiúsculas e minúsculas da parte do Nome distinto, excluindo quaisquer curingas.

Se dois DNs forem iguais em todos os aspectos, exceto por seus valores de DC, as mesmas regras de correspondência se aplicarão para OUs, exceto que em valores de DC, o DC mais à esquerda é o nível mais baixo (mais específico) e a ordenação de comparação difere em conformidade.

Exibindo registros de autenticação de canal

Para exibir registros de autenticação de canal, use o comando MQSC **DISPLAY CHLAUTH** ou o comando PCF **Inquire Channel Authentication Records**. É possível escolher para retornar todos os registros que correspondam ao nome de canal fornecido ou é possível escolher um correspondência explícita. A correspondência explícita informa qual registro de autenticação canal seria usado se um canal tentasse fazer uma conexão a partir de um endereço IP específico, a partir de um gerenciador de filas específico ou usando um ID do usuário específico e, opcionalmente, apresentando um certificado pessoal de Secure Sockets Layer/Segurança da Camada de Transporte contendo um Nome distinto especificado.

Conceitos relacionados

"Segurança para o Sistema de Mensagens Remoto" na página 105

Esta seção trata dos aspectos do sistema de mensagens remoto de segurança.

Interação de CHLAUTH e CONNAUTH

Como os registros de autenticação de canal (CHLAUTH) e a autenticação de conexão (CONNAUTH) interagem no IBM MQ, no caso de uma conversa única em um canal.

Tipos diferentes de ligações

O IBM MQ suporta dois métodos para que um aplicativo se conecte:

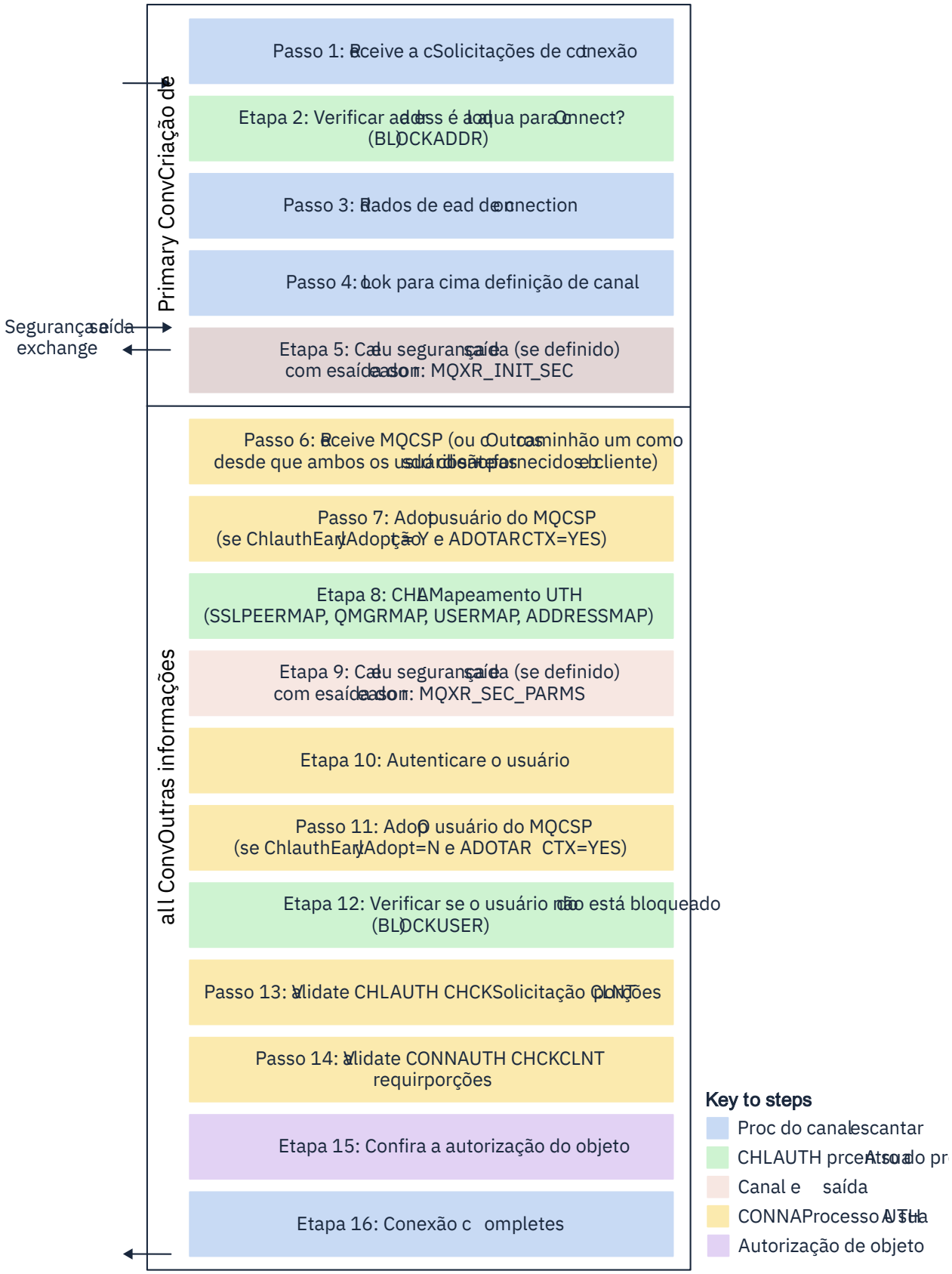
Ligações locais
Aplica-se quando o aplicativo e o gerenciador de filas estão na mesma imagem operacional. O CHLAUTH não é relevante para esse tipo de conexão de aplicativo.

Ligações do Cliente
Aplica-se quando o aplicativo e o gerenciador de filas usam a rede para se comunicar. O aplicativo e o gerenciador de filas podem estar em execução na mesma máquina ou podem estar em máquinas diferentes. No IBM MQ, uma conexão do cliente é manipulada na forma de um canal de conexão do servidor (SVRCONN) e, nessa situação, tanto CONNAUTH quanto CHLAUTH são aplicáveis.

Etapas de ligação da extremidade de recebimento de um canal

Quando um aplicativo se conecta a um gerenciador de filas, uma quantia substancial de verificação é feita para assegurar que ambas as extremidades do canal entendam o que é suportado pela outra extremidade. A extremidade de recebimento do canal faz uma verificação extra, envolvendo CHLAUTH e CONNAUTH, para assegurar que o cliente tenha permissão para se conectar e esse processo também pode incluir uma saída de segurança, já que isso pode afetar o resultado. Essa fase de conexão de canal também é referida como *fase de ligação*.

O diagrama a seguir lista as etapas pelas quais um canal SVRCONN passa quando o encerramento do servidor (no Gerenciador de Filas) inicia:



Etapa 1: Receber uma solicitação de conexão
O inicializador ou o listener de canais recebe uma solicitação de conexão de algum lugar na rede.

Etapa 2: O endereço tem permissão para se conectar?
Antes que quaisquer dados sejam lidos, o IBM MQ verifica o endereço IP do parceiro com relação às regras do CHLAUTH, para ver se o endereço está na regra *BLOCKADDR*. Se o endereço não for localizado e, portanto, não estiver bloqueado, o fluxo continuará para a próxima etapa.

Etapa 3: Ler dados do canal
O IBM MQ agora lê os dados em um buffer e começa a processar as informações enviadas.

Etapa 4: Consultar a definição de canal
No primeiro fluxo de dados, IBM MQ envia, entre outras coisas, o nome do canal que a extremidade de envio está tentando iniciar. O gerenciador de filas de recebimento pode, então, consultar a definição de canal, que possui todas as configurações que são especificadas para o canal.

Etapa 5: Saída de segurança de chamada (se definida)
Se o canal tiver uma saída de segurança (SCYEXIT) definida, ela será chamada com o motivo da saída (MQCXP**ExitReason**) configure para MQXR_INIT_SEC.

Etapa 6: Receber MQCSP
Se necessário, construa um se as credenciais de autenticação fornecidas pelo cliente.

Se o cliente for um aplicativo Java ou JMS executando em modo de compatibilidade, o cliente não passará uma estrutura MQCSP para o Gerenciador de Filas. Em vez disso, se o aplicativo tiver fornecido um ID do usuário e uma senha, uma estrutura do MQCSP será construída aqui.

Etapa 7: adotar usuário MQCSP (se `ChlauthEarlyAdopt` for *Y* e ADOPTCTX=YES)
As credenciais fornecidas pelo cliente são autenticadas.

Se CONNAUTH estiver usando o LDAP para mapear um nome distinto declarado para um ID de usuário curto, o mapeamento acontecerá nessa etapa.

Se a autenticação for bem-sucedida, o ID do usuário será adotado pelo canal e usado pela etapa de mapeamento CHLAUTH.

Nota: A partir do IBM MQ 9.0.4, o parâmetro **`ChlauthEarlyAdopt=`** *Y* é automaticamente incluído na sub-rotina de canais do arquivo qm.ini para novos gerenciadores de filas.

Etapa 8: Mapeamento CHLAUTH
O cache CHLAUTH é inspecionado novamente para procurar as regras de mapeamento *SSLPEERMAP*, *USERMAP*, *QMGRMAP* e *ADDRESSMAP*.

A regra que corresponde ao canal de entrada mais especificamente é usada. Se a regra tiver **USERSRC***(CHANNEL)* ou *(MAP)*, o canal continua em vinculado.

Se as regras CHLAUTH forem avaliadas para uma regra com **USERSRC***(NOACCESS)*, o aplicativo será bloqueado de se conectar ao canal, a menos que as credenciais sejam subsequentemente substituídas por credenciais válidas na Etapa 9.

Etapa 9: Chamar saída de segurança (se definida)
Se o canal tiver uma saída de segurança (SCYEXIT) definida, ela será chamada com o motivo da saída (MQCXP**ExitReason**) configurado para MQXR_SEC_PARMS.

Um ponteiro para MQCSP estará presente no campo **SecurityParms** da estrutura MQCXP.

A estrutura MQCSP tem ponteiros para o ID do usuário (MQCSP.**CSPUserIdPtr**) e senha (MQCSP.**CSPPasswordPtr**). V 9.4.0 De IBM MQ 9.3.4, a estrutura MQCSP também contém um ponteiro para o token de autenticação (MQCSP.**TokenPtr**).

É possível alterar o ID do usuário e a senhae o token de autenticação na saída. O exemplo a seguir mostra como uma saída de segurança imprimiria os valores de ID do usuário e senha para um log de auditoria:

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

```
        pMQCXP -> SecurityParms -> CSPUserIdLength,
        pMQCXP -> SecurityParms -> CSPUserIdPtr,
        pMQCXP -> SecurityParms -> CSPPasswordLength,
        pMQCXP -> SecurityParms -> CSPPasswordPtr);
```

A saída pode dizer IBM MQ para fechar o canal, retornando *MQXCC_CLOSE_CHANNEL* no MQCXP.campo **Exitresponse**. Caso contrário, o processamento do canal continuará com a fase de autenticação de conexão.

Nota: Se o usuário declarado for mudado pela saída de segurança, as regras de mapeamento de CHLAUTH não serão reaplicadas ao novo usuário

Etapa 10: Autenticar o usuário

A fase de autenticação ocorre quando CONNAUTH é ativado no gerenciador de filas.

Para verificar isso, emita o comando 'DISPLAY QMGR CONNAUTH' do MQSC.

z/OS O exemplo a seguir mostra a saída do comando **DISPLAY QMGR CONNAUTH** de um gerenciador de filas em execução no IBM MQ for z/OS.

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

Multi O exemplo a seguir mostra a saída do comando '**DISPLAY QMGR CONNAUTH**' de um gerenciador de filas em execução no IBM MQ for Multiplatforms.

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

O valor CONNAUTH é o nome de um objeto **AUTHINFO** IBM MQ.

Como a autenticação do sistema operacional (**AUTHTYPE**(*IDPWOS*)) é válida em ambos IBM MQ for Multiplatforms e IBM MQ for z/OS, os exemplos usam a autenticação do sistema operacional.

z/OS O exemplo a seguir mostra o objeto AUTHINFO padrão com **AUTHTYPE**(*IDPWOS*) de um gerenciador de filas em execução no IBM MQ for z/OS.

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

Multi O exemplo a seguir mostra o objeto AUTHINFO padrão com **AUTHTYPE**(*IDPWOS*) de um gerenciador de filas em execução no IBM MQ for Multiplatforms.

```
 1 : display authinfo(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
AMQ8566: Display authentication information details.
   AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
   AUTHTYPE(IDPWOS)                        ADOPTCTX(NO)
   DESCR( )                                CHCKCLNT(REQDADM)
   CHCKLOCL(OPTIONAL)                      FAILDLAY(1)
   ALTDATE(2015-06-08)                     ALTTIME(16.35.16)
```

O objeto AUTHINFO TYPE (IDPWOS) possui um atributo chamado CHCKCLNT.. Se o valor for alterado para *REQUIRED* , todos os aplicativos clientes deverão fornecer credenciais válidas.

Se o usuário foi autenticado na Etapa 7, outra verificação de autenticação não será executada, a menos que:

- O ID do usuário e a senha, ou o token de autenticação, no campo SecurityParms da estrutura MQCXP foi alterado por uma saída de segurança na Etapa 9.
- O aplicativo cliente foi conectado com opções solicitando a funcionalidade reconectável.

Etapa 11: adotar o contexto do usuário MQCSP (Se `ChlauthEarlyAdopt=`*N* e ADOPTCTX=YES)
É possível configurar o atributo ADOPTCTX, que controla se o canal é executado sob MCAUSER ou o ID do usuário que o aplicativo forneceu.

Se o ID do usuário declarado no MQCSP ou no campo **`SecurityParms`** da estrutura MQXCP tiver sido autenticado com êxito e **ADOPTCTX** for *YES*, o contexto do usuário resultante das etapas 7 e 8 será adotado como o contexto a ser usado para esse aplicativo, a menos que o ID do usuário e a senha, ou o token de autenticação no campo **`SecurityParms`** da estrutura MQCXP tenha sido alterado por uma saída de segurança na etapa 9.

Esse ID do usuário declarado é o ID do usuário que é verificado para autorização para usar os recursos do IBM MQ.

Por exemplo, você não tem um conjunto MCAUSER no canal SVRCONN e seu cliente está em execução no '`johndoe`' em sua máquina Linux. Seu aplicativo especifica o usuário '`fred`' no MQCSP, assim, o canal começa a ser executado com '`johndoe`' como o MCAUSER ativo. Após a verificação do CONNAUTH, o usuário '`fred`' é adotado, e o canal é executado com '`fred`' como o MCAUSER ativo.

Etapa 12: Verificar se o usuário não está bloqueado (BLOCKUSER)
Se a verificação de CONNAUTH for bem-sucedida, o cache CHLAUTH será inspecionado novamente para verificar se o MCAUSER ativo está bloqueado por uma regra *BLOCKUSER* . Se o usuário estiver bloqueado, o canal será encerrado.

Etapa 13: Validar os Requisitos de CHLAUTH CHCKCLNT
Se a regra CHLAUTH selecionada na etapa 8 especificar adicionalmente um valor CHCKCLNT de REQUIRED ou REQDADM, a validação será feita para assegurar que um ID do usuário CONNAUTH válido foi fornecido para atender ao requisito.

- Se CHCKCLNT (REQUIRED) estiver configurado, um usuário deverá ter sido autenticado na etapa 7 ou 10.. Caso contrário, a conexão será rejeitada
- Se CHCKCLNT (REQDADM) estiver configurado, um usuário deverá ter sido autenticado na etapa 7 ou 10 se for determinado que essa conexão é privilegiada Caso contrário, a conexão será rejeitada
- Se CHCKCLNT (ASQMGR) for configurado, esta etapa será ignorada.

Notas:

1. Se CHCKCLNT (REQUIRED) ou CHCKCLNT (REQDADM) estiver configurado, mas CONNAUTH não estiver ativado no gerenciador de filas, a conexão falhará com um código de retorno MQRC_SECURITY_ERROR (2063) devido ao conflito na configuração..
2. O usuário não é autenticado novamente nesta etapa

Etapa 14: Validar os requisitos de CONNAUTH CHCKCLNT.
A fase de autenticação ocorre quando CONNAUTH é ativado no gerenciador de filas.

O valor CONNAUTH CHCKCLNT é verificado para determinar quais requisitos são configurados para conexões de entrada:

- Se CHCKCLNT (NONE) for configurado, esta etapa será ignorada
- Se CHCKCLNT (OPTIONAL) for configurado, esta etapa será ignorada.
- Se CHCKCLNT (REQUIRED) for configurado, um usuário deverá ter sido autenticado na etapa 7 ou 10.. Caso contrário, a conexão será rejeitada
- Se CHCKCLNT (REQDADM) estiver configurado, um usuário deverá ter sido autenticado na etapa 7 ou 10 se for determinado que essa conexão é privilegiada Caso contrário, a conexão será rejeitada

Nota: O usuário não é autenticado novamente nesta etapa

Multi **Etapa 15: Verificar autorização de objeto**
Uma verificação é feita para assegurar que o MCAUSER ativo tenha a autoridade apropriada para se conectar ao gerenciador de filas.

ALW Consulte Gerenciador de autoridade de objeto para obter mais informações.

IBM i Consulte “Gerenciador de autoridade de objeto no IBM i” na página 164, para mais informações.

Etapa 16: A conexão é concluída
Se as etapas anteriores forem concluídas com sucesso, a conexão estará concluída.

Conceitos relacionados
CONNAUTH
Um gerenciador de filas pode ser configurado para autenticar credenciais que são fornecidas por um aplicativo quando ele se conecta..

Referências relacionadas
SET CHLAUTH
ALTER AUTHINFO

Resolvendo problemas de acesso CHLAUTH
Etapas e exemplos para resolver determinados problemas de acesso ao usar registros de autenticação de canal (CHLAUTH).

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver AdministrandoIBM MQ usando comandos MQSC .

Sobre esta tarefa

Há três regras padrão para o processamento de CHLAUTH:

- NO ACCESS para todos os canais por quaisquer usuários `MQ-admin*`
- NO ACCESS to all SYSTEM.* canais por todos os usuários
- Acesso ALLOW ao canal SYSTEM.ADMIN.SVRCONN (usuários não `MQ-admin`)

As duas primeiras regras bloqueiam o acesso a todos os canais. A terceira regra é mais específica e, portanto, tem precedência sobre as outras duas, caso o canal seja o canal SYSTEM.ADMIN.SVRCONN, permitindo, portanto, o acesso nesse canal.

As regras CHLAUTH são usadas para determinar se um canal pode ser iniciado e permitem o mapeamento, por meio de MCAUSER, para outro ID do usuário. Se o canal não puder ser iniciado, os erros a seguir geralmente ocorrerão:

- `RC 2035 MQRC_NOT_AUTHORIZED`
- `RC 2059 MQRC_Q_MGR_NOT_AVAILABLE`
- `AMQ4036 Acesso não permitido`
- `AMQ9776: O canal foi bloqueado pelo ID do usuário`
- `AMQ9777: O canal foi bloqueado`
- `MQJE001: Ocorreu uma MQException: Código de Conclusão 2, Razão 2035`
- `MQJE036: O gerenciador de filas rejeitou uma tentativa de conexão`

É necessário bloquear o acesso estritamente e, em seguida, incluir mais regras CHLAUTH para controlar quem pode acessar e iniciar os canais.

Como uma medida temporária, e para solucionar os erros listados, conclua qualquer uma das etapas a seguir:

Procedimento

- **Desativar regras CHLAUTH**

  Como uma medida provisória e também para solucionar os erros acima, é possível desativar as regras CHLAUTH. As regras podem ser reativadas a qualquer momento e se a desativação das regras CHLAUTH resolver o problema de conexão, você sabe que essa foi a causa.

  Para desativar as regras CHLAUTH, execute o comando MQSC a seguir:

```
ALTER QMGR CHLAUTH (DISABLED)
```

  Observe que também é possível configurar CHLAUTH como *WARN*, que permite o acesso e registra o resultado da regra.

- **Modificar ou remover regras CHLAUTH**

  Também é possível excluir ou modificar uma ou mais regras CHLAUTH, que causa seu problema.

  Para modificar uma regra CHLAUTH, utilize o comando SET CHLAUTH com ACTION (REPLACE). Por exemplo, para modificar a regra padrão que não causa acesso a todos os canais por nenhum usuário do `MQ-admin` para WARN, em vez de ser bloqueado, execute o comando MQSC a seguir:

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

  Para excluir uma regra CHLAUTH, utilize o comando SET CHLAUTH com a ACTION (REMOVE). Por exemplo, para excluir a regra padrão que não causa acesso a todos os canais por nenhum usuário do `MQ-admin` , execute o comando MQSC a seguir:

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

- **Acesso de teste usando MATCH (RUNCHECK)**

  É possível testar o resultado de suas regras CHLAUTH, usando a opção MATCH (*RUNCHECK*) da regra CHLAUTH. A opção **MATCH** (*RUNCHECK*) retorna o registro que é correspondido por um canal de entrada específico no tempo de execução, quando esse canal se conecta a esse gerenciador de filas. Deve-se fornecer:

  – O nome do canal
  – Atributo ADDRESS
  – Atributo SSLPEER, apenas se o canal de entrada usar SSL ou TLS
  – QMNAME, se o canal de entrada for um canal do gerenciador de filas ou
  – Atributo CLNTUSER, se o canal de entrada for um canal do cliente

  O exemplo a seguir executa um comando MQSC para verificar qual regra CHLAUTH, com as regras padrão em vigor, resulta em um `MQ-admin` usuário `johndoe` acessando um canal denominado `CHAN1`:

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

AMQ8878: Display channel authentication record details.
CHLAUTH(*) TYPE(BLOCKUSER)
USERLIST(*MQADMIN)
```

  Para o usuário `johndoe`, o canal não é executado e o usuário será bloqueado devido à regra BLOCKUSER para usuários *MQADMIN.

O exemplo a seguir executa um comando MQSC para verificar qual regra CHLAUTH, com as regras padrão em vigor, resulta no usuário `alice` que não é um usuário `MQ-admin` , acessando um canal denominado CHAN1:

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

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

Para o usuário `alice`, o canal é executado e transmite `alice` como o MCAUSER. O MCAUSER é o ID do usuário usado para verificar as autoridades de objetos do IBM MQ.

Referências relacionadas

SET CHLAUTH

DISPLAYCHLAUTH

Criando novas regras CHLAUTH para usuários

Alguns cenários comuns para usuários e regras CHLAUTH de exemplo para realizá-los.

Antes de começar

Nota: As etapas nesta tarefa requerem que você execute comandos do MQSC A maneira como você faz isso varia de acordo com a plataforma. Ver AdministrandoIBM MQ usando comandos MQSC .

Sobre esta tarefa

Há três regras padrão para o processamento de CHLAUTH:

- NO ACCESS para todos os canais por quaisquer usuários `MQ-admin*`
- NO ACCESS to all SYSTEM.* canais por todos os usuários
- Acesso ALLOW ao canal SYSTEM.ADMIN.SVRCONN (usuários não `MQ-admin`)

As duas primeiras regras bloqueiam o acesso a todos os canais. A terceira regra é mais específica e, portanto, tem precedência sobre as outras duas, caso o canal seja o canal SYSTEM.ADMIN.SVRCONN, permitindo, portanto, o acesso nesse canal.

Para criar novas regras CHLAUTH para usuários, configure um ou mais dos seguintes cenários.

Procedimento

- **Controle o acesso para usuários específicos do `MQ-admin`**
  a) Configure um canal de conexão do servidor que deve ser usado exclusivamente para uma perspectiva administrativa, ou seja, para conectar a partir do IBM MQ Explorer.

  Você tem um canal específico para esse uso, um ou mais endereços IP definidos de onde deseja que as conexões sejam aceitas e acesso bloqueado para o ID `'mqm'`, caso a conexão não seja de um dos endereços IP especificados.

  b) Crie um canal SVRCONN para IBM MQ Explorer e `MQ-admin` usuários chamados `ADMIN.CHAN`.

  Execute o seguinte comando MQSC:

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

  c) Para teste, assegure-se de que tenha um usuário definido que esteja no grupo `MQ-admin` e um que não esteja.

  Para este cenário, `mqadm` está no grupo `MQ-admin` e `alice` não está.

  d) Confirme se as regras CHLAUTH padrão estão em vigor..

  e) Inclua três regras para permitir que um usuário específico acesse ADMIN.CHAN como `MQ-admin` por meio de determinados endereços IP:

– Configure NOACCESS por meio de qualquer endereço
– Configure BLOCKUSER para esse canal para bloquear apenas o usuário nobody, que substitui o *MQADMIN BLOCKUSER
– Permita acesso ALLOW ao usuário mqadm em uma sub-rede de endereços específica e MAP para autoridade de usuário mqadm

Para fazer isso, execute os comandos MQSC a seguir:

```
SET CHLAUTH (ADMIN.CHAN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
SET CHLAUTH('ADMIN.CHAN') TYPE(BLOCKUSER) +
DESCR('Rule to override *MQADMIN blockuser on this channel') +
USERLIST('nobody') ACTION(replace)
SET CHLAUTH('ADMIN.CHAN') TYPE(USERMAP) +
CLNTUSER('mqadm') USERSRC(MAP) MCAUSER('mqadm') +
ADDRESS('192.168.1.*') +
DESCR('Allow mqadm as mqadm on local subnet') ACTION(ADD)
```

Neste ponto, o usuário mqadm pode acessar e iniciar o canal ADMIN.CHAN, por meio de um intervalo de endereço IP especificado.

f) Opcional: É possível executar o comando MQSC MATCH (RUNCHECK) a qualquer momento para ver os resultados de cada um destes comandos:

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

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

Neste ponto, apenas os usuários que têm um registro CHLAUTH têm permissão para acessar usando o ADMIN.CHAN.

- **Controle o acesso para um usuário específico e IBM MQ aplicativo cliente**

  Para esse cenário, as regras CHLAUTH padrão são adequadas, assumindo que a autoridade IBM MQ deve ser configurada para um usuário específico, para fornecer a autoridade IBM MQ correta (usando setmqaut).

  Neste cenário, as autoridades são configuradas para um usuário mqapp1, que não é um usuário MQ-admin.

  a) Use o seguinte comando MQSC para criar um canal SVRCONN, APP1.CHAN, a ser usado por um aplicativo específico e um usuário específico

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

  b) Com as regras CHLAUTH padrão em vigor, o usuário mqapp1 pode iniciar o canal APP1.CHAN.

  O ID do usuário proveniente do aplicativo cliente do IBM MQ é usado para verificação de autoridade de objeto do IBM MQ. Nesse caso, supondo que o usuário mqapp1 esteja executando o aplicativo cliente IBM MQ , ele será usado na verificação de autoridade de objeto IBM MQ . Portanto, se mqapp1 tiver acesso aos objetos do IBM MQ de que o aplicativo precisa, tudo estará normal, caso contrário, você obterá erros de autoridade.

  É possível aumentar a segurança ainda mais ao criar regras CHLAUTH específicas para o ID do usuário mqapp1, no entanto, sob as regras padrão, nenhum membro do grupo MQ-admin pode acessar esse canal.

Execute os comandos MQSC a seguir:

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

- **Controle o acesso para um usuário específico usando o nome distinto (DN) do certificado desse usuário**

  Para este cenário, o usuário deve ter um certificado que seja encaminhado para o gerenciador de filas. O DN é, então, correspondido com a configuração SSLPEER da regra CHLAUTH e o SSLPEER pode utilizar caracteres curinga.

  Se correspondido, o usuário também poderá ser mapeado para um MCAUSER diferente para propósitos de verificação das autoridades de objeto do IBM MQ. O mapeamento do MCAUSER pode minimizar o número de usuários que precisam ser gerenciados no gerenciador de autoridade de objeto (OAM) do IBM MQ.

  a) Você tem um canal TLS com certificados em uso e requer regras para:

  – Bloquear todos os usuários para um canal específico
  – Permitir apenas usuários com um determinado SSLPEER que usam o cliente desse usuário para acesso do IBM MQ OAM.

  Execute os comandos MQSC a seguir:

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

  O ID do usuário do cliente que está se conectando no canal é usado para a autoridade do IBM MQ OAM de objetos do IBM MQ, portanto, o ID do usuário deve ter autoridades apropriadas do IBM MQ.

  b) Opcional: Mapeie para um ID do usuário IBM MQ diferente.

  Execute novamente o comando MQSC anterior, substituindo `USERSRC(MAP) MCAUSER('mquser1')` por `USERSRC(CHANNEL)`..

- **Mapear um usuário específico para o `mqm` usuário**

  Essa é uma adição ou modificação para Controlar o acesso para usuários MQ-admin específicos

  Use comandos MQSC para incluir a regra CHLAUTH a seguir para mapear usuários específicos para o usuário `mqm` , ou um ID do usuário `MQ-admin` , que tenha a configuração de autoridade de objeto IBM MQ no OAM IBM MQ

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

  Isso permite e mapeia o usuário `johndoe` para o usuário `mqm` para o canal ADMIN.CHAN específico.

Conceitos relacionados

“Criando novas regras CHLAUTH para canais” na página 70

Para ajudar a criar suas próprias regras CHLAUTH, aqui estão alguns cenários comuns para os canais, e, por exemplo, regras CHLAUTH para realizar isso.

Tarefas relacionadas

"Resolvendo problemas de acesso CHLAUTH" na página 65
Etapas e exemplos para resolver determinados problemas de acesso ao usar registros de autenticação de canal (CHLAUTH).

Referências relacionadas

SET CHLAUTH

DISPLAYCHLAUTH

Criando novas regras CHLAUTH para canais

Para ajudar a criar suas próprias regras CHLAUTH, aqui estão alguns cenários comuns para os canais, e, por exemplo, regras CHLAUTH para realizar isso.

Este tópico contém os cenários a seguir:

Permitir acesso a um determinado canal por meio de um intervalo de endereços IP específico

Para este cenário, você deseja:

- Configurar Nenhum acesso ao canal de qualquer lugar
- Permitir acesso por meio de um endereço IP ou de intervalo de endereços específico

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

Isso permite que apenas o canal APP2.CHAN seja iniciado quando a condição é proveniente de um intervalo de endereço IP específico determinado.

O usuário que se conecta como MCAUSER é mapeado para `mqapp2` e, portanto, obtém a autoridade OAM do IBM MQ para esse usuário.

Para um canal específico, bloqueie todos os usuários, mas permita que usuários específicos se conectem.

Há três regras padrão para o processamento de CHLAUTH:

- NO ACCESS para todos os canais por quaisquer usuários `MQ-admin*`
- NO ACCESS to all SYSTEM.* canais por todos os usuários
- Acesso ALLOW ao canal SYSTEM.ADMIN.SVRCONN (usuários não `MQ-admin`)

As duas primeiras regras bloqueiam o acesso a todos os canais. A terceira regra é mais específica e, portanto, tem precedência sobre as outras duas, caso o canal seja o canal SYSTEM.ADMIN.SVRCONN, permitindo, portanto, o acesso nesse canal.

Para esse cenário, o acesso ao canal MY.SVRCONN possui as regras CHLAUTH padrão no lugar.

É necessário incluir o seguinte:

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

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

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

Essa primeira parte do código impede que qualquer pessoa se conecte a MY.SVRCONN e, em seguida, o código permite que apenas o canal MY.SVRCONN seja iniciado quando a conexão é proveniente do ID do usuário específico `johndoe`.

O usuário que se conecta no canal `johndoe` é usado para a autoridade do OAM do IBM MQ de objetos do IBM MQ. Portanto, o ID do usuário deve ter as autoridades apropriadas do IBM MQ.

Se desejar, será possível mapear para um ID do usuário do IBM MQ diferente usando:

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

em vez de `USERSRC(CHANNEL)`.

Usando CHLAUTH para canais receptores e emissores

É possível usar regras CHLAUTH para incluir segurança extra nos canais receptores e emissores, para restringir o acesso ao canal receptor. Observe que, se você estiver incluindo ou fazendo mudanças nas regras CHLAUTH, as regras CHLAUTH atualizadas serão aplicadas apenas ao iniciar o canal, portanto, se os canais já estiverem em execução, será necessário pará-los e reiniciá-los, para que as atualizações CHLAUTH sejam aplicadas.

As regras CHLAUTH podem ser usadas em qualquer canal, mas há algumas restrições. Por exemplo, as regras USERMAP se aplicam apenas a canais SVRCONN.

Este exemplo permite uma conexão somente por meio de um endereço IP específico, para iniciar o canal TO.MYSVR1:

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

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

Este exemplo permite a conexão apenas de um gerenciador de filas específico:

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

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

Tarefas relacionadas

“Resolvendo problemas de acesso CHLAUTH” na página 65
Etapas e exemplos para resolver determinados problemas de acesso ao usar registros de autenticação de canal (CHLAUTH).

“Criando novas regras CHLAUTH para usuários” na página 67
Alguns cenários comuns para usuários e regras CHLAUTH de exemplo para realizá-los.

Referências relacionadas
SET CHLAUTH
DISPLAYCHLAUTH

Criando uma regra de backstop CHLAUTH
Ao pensar sobre o controle de conexões de entrada no seu gerenciador de filas, você tem duas opções. Ou é possível tentar listar todas as conexões que não são permitidas ou é possível começar dizendo que todas as conexões não são permitidas e tentar listar todas as conexões permitidas. Essa segunda opção é descrita aqui.

Sobre esta tarefa

O motivo para usar a segunda opção é que, se você tentar listar todas as conexões que não são permitidas e, portanto, tudo que não for listado será permitido, a ausência de uma conexão na lista fará com que uma conexão que não deveria ter sido permitida consiga de se conectar, causando uma violação de segurança em potencial.

Inversamente, se, em vez disso, você começar dizendo que toda conexão não é permitida e listar aquelas que são, a ausência de uma conexão nessa lista não será uma violação de segurança. Se a sua empresa requerer que conexões adicionais sejam incluídas, essa será uma tarefa relativamente simples e não haverá nenhuma violação de segurança potencial.

Primeiro, crie uma regra de *backstop*, que é uma regra que captura todas as conexões que não são correspondidas por regras mais específicas. Essa regra tem o efeito de impedir qualquer conexão remota de se conectar ao seu gerenciador de filas.

No entanto, se você estiver preocupado com essa abordagem, será possível configurar a regra de *backstop* no modo de aviso. Veja a etapa “2” na página 72

Procedimento

1. Para criar uma regra de backstop que impeça as conexões remotas de se conectarem ao seu gerenciador de filas, emita o comando a seguir:

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

   Agora que você fechou a porta em todas as conexões remotas, é possível começar a estabelecer regras mais específicas para permitir a entrada de determinadas conexões. Por exemplo:

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

2. Se você desejar criar a regra de backstop no modo de aviso, emita o comando a seguir:

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

   Agora é possível continuar e fazer todas as suas regras positivas. Quando você achar que criou todas as regras necessárias, ative os eventos do canal emitindo o comando a seguir:

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

   e monitore a fila SYSTEM.ADMIN.CHANNEL.EVENT para eventos com **Reason** configurado como MQRC_CHANNEL_BLOCKED_WARNING.

   Esses eventos detalham as conexões que corresponderam a sua regra de backstop, mas como o comando está em execução no modo de aviso, elas não estão bloqueadas no momento.

Revise cada um desses eventos e determine se essa conexão deve ter uma regra positiva estabelecida para que ela seja permitida ou se ela foi correspondida corretamente com relação à regra de *backstop*. É possível executar nesse modo, revisando os eventos conforme eles são criados, até que você esteja satisfeito por ter visto todos os canais de entrada e por ter regras positivas adequadas estabelecidas para todos eles.

Neste ponto, é possível mudar a regra de *backstop* para começar, de fato, o bloqueio das conexões correspondentes emitindo o comando a seguir:

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

Criando um administrador IBM MQ não privilegiado
Como criar um administrador não privilegiado do IBM MQ usando o CHLAUTH.

Sobre esta tarefa

No contexto dessa tarefa, os termos:

usuário privilegiado
Significa um usuário que tem autorização para executar uma operação sem que o acesso tenha sido explicitamente concedido a ele para executar essa operação. Os usuários no grupo mqm são exemplos desses usuários privilegiados.

Administrador do IBM MQ
Significa um usuário que precisa emitir comandos administrativos em IBM MQ, como **DEFINE QLOCAL** ou **START CHANNEL**.

As etapas a seguir criam um administrador não privilegiado do IBM MQ.

Procedimento

1. Crie um ID do usuário na máquina do gerenciador de filas usando os comandos apropriados para a plataforma ou plataformas que sua empresa usa.

   O nome do usuário `alice` é usado neste nesse exemplo.
2. Conceda a esse novo usuário a autoridade para emitir todos os comandos administrativos do IBM MQ executando o procedimento a seguir:
   a) Inicie o IBM MQ Explorer usando um usuário privilegiado.
   b) Navegue para o *Assistente baseado em regra* selecionando o gerenciador de filas apropriado, em seguida, `Autoridades de objeto` e `Incluir autoridades baseadas em regra`.
   c) No painel de assistente que aparece, insira o ID do usuário criado na primeira etapa ou, se preferir trabalhar com grupos, insira o nome do grupo para o usuário ou conjunto de usuários que você deseja transformar em administradores não privilegiados do IBM MQ.
   d) Configure o assistente para acesso administrativo total.
   e) Se você deseja permitir que o administrador não privilegiado do IBM MQ seja capaz de navegar pelas mensagens nas filas, marque também essa caixa de seleção.
   f) Revise os comandos no painel de visualização na parte inferior do assistente.

      É possível cortar e colar esses comandos para construir seus próprios scripts.

      Uma razão pela qual você pode preferir fazer isso com seu próprio script é reduzir a quantidade de acesso que você concede a esse usuário. Talvez, em vez de conceder acesso a todos os objetos, seja preferível conceder acesso apenas a um determinado grupo de objetos.

      Pressionar **OK** no assistente emite os comandos conforme eles são mostrados.
   g) Será necessário configurar algumas regras CHLAUTH para permitir o acesso remoto para esse ID do usuário se o requisito para um administrador não privilegiado do IBM MQ também for para acesso remoto.

Assumindo que sua empresa esteja usando a orientação no “Criando uma regra de backstop CHLAUTH” na página 72, tudo o que você precisa fazer é incluir uma regra de ativação.

A regra que você cria depende de como escolhe autenticar seus administradores remotos do IBM MQ.

Se você estiver usando a autenticação TCP/IP fraca, será possível configurar uma regra CHLAUTH semelhante ao seguinte:

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

9. Se você estiver usando a autenticação TLS, será possível configurar uma regra CHLAUTH semelhante à seguinte:

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

Agora, quando um usuário se conecta no `admin-channel-name` (e corresponde às regras CHLAUTH), ele é capaz de emitir comandos sob o ID do usuário `alice` no gerenciador de filas e, assim, o acesso remoto privilegiado não é necessário.

Autenticação de conexão

A autenticação de conexão permite que os aplicativos forneçam as credenciais de autenticação quando se conectam a um gerenciador de filas O gerenciador de filas valida as credenciais O ID do usuário fornecido nas credenciais também pode ser adotado para uso em verificações de autorização para os recursos que o aplicativo acessa..

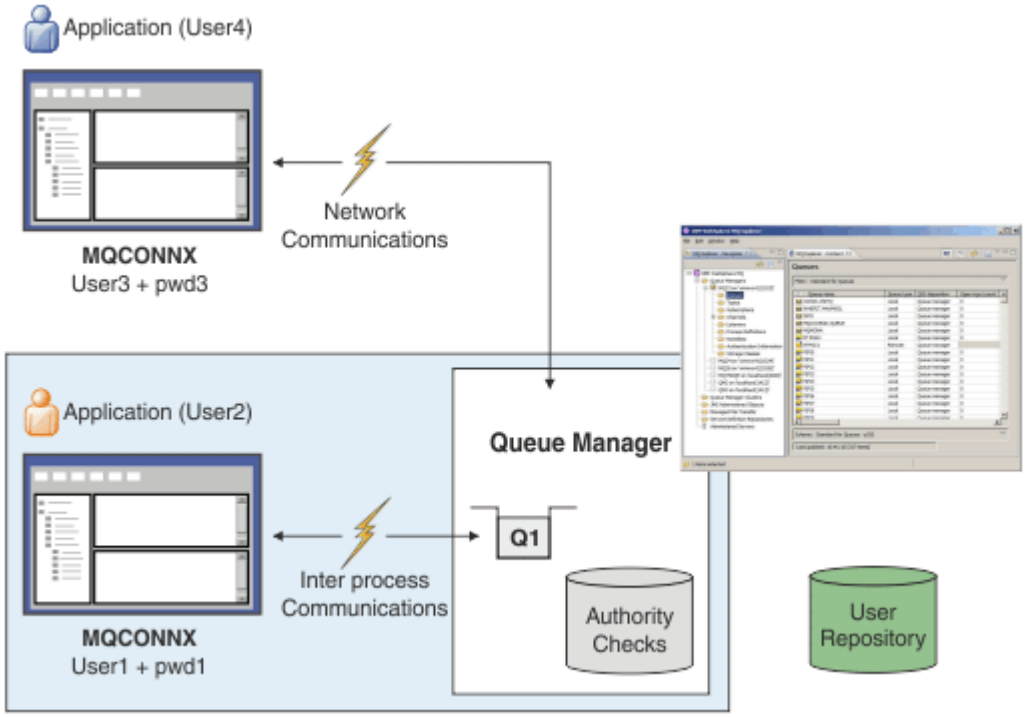
Os aplicativos podem fornecer um ID do usuário e senha para autenticação quando eles se conectam a um gerenciador de filas.

V 9.4.0 No IBM MQ 9.3.4, os aplicativos IBM MQ client também podem fornecer um token de autenticação como um método alternativo de autenticação.

O gerenciador de filas pode ser configurado para validar as credenciais fornecidas pelo aplicativo.

Um ID do usuário e uma senha fornecidos por um aplicativo são verificados usando o repositório do usuário na configuração do gerenciador de filas. Para obter mais informações sobre o repositório que é usado para verificar IDs do usuário e senhas, consulte Repositórios do usuário

V 9.4.0 Os tokens de autenticação são validados usando os certificados e as chaves simétricas no keystore de autenticação de token do gerenciador de filas para validar a assinatura do token. Para obter mais informações sobre como autenticar usuários com tokens de autenticação, consulte “Trabalhando com tokens de autenticação.” na página 330



No diagrama, dois aplicativos estão fazendo conexões com um gerenciador de filas, um aplicativo como um cliente e um usando ligações locais. Os aplicativos podem usar várias APIs para se conectarem ao gerenciador de filas, mas todos têm a capacidade de fornecer um ID do usuário e uma senha. O ID do usuário sob o qual o aplicativo está em execução, `User2` e `User4` no diagrama, que é o ID do usuário do sistema operacional usual apresentado para o IBM MQ pode ser diferente do ID do usuário fornecido pelo aplicativo, `User1` e `User3`.

O gerenciador de filas recebe comandos de configuração (no diagrama, o IBM MQ Explorer está sendo usado) e gerencia a abertura de recursos e verifica a autoridade para acessar esses recursos. Há muitos recursos diferentes no IBM MQ que um aplicativo pode requerer autoridade para acessar. O diagrama ilustra como abrir uma fila para saída, mas os mesmos princípios também se aplicam a outros recursos.

Conceitos relacionados

“Autenticação de conexão: configuração” na página 75
Um gerenciador de filas pode ser configurado para autenticar credenciais que são fornecidas por um aplicativo quando ele se conecta..

“Autenticação de conexão: Mudanças no aplicativo” na página 80

“Autenticação de conexão: Repositórios do usuário” na página 81
Para cada um de seus gerenciadores de filas, é possível escolher diferentes tipos de objeto de informações sobre autenticação para autenticar os IDs do usuário e senhas.

Autenticação de conexão: configuração

Um gerenciador de filas pode ser configurado para autenticar credenciais que são fornecidas por um aplicativo quando ele se conecta..

Ativando a autenticação de conexão em um gerenciador de filas

Em um objeto de gerenciador de filas, o atributo **CONNAUTH** pode ser definido para o nome de um objeto de informações sobre autenticação (AUTHINFO). O atributo **AUTHTYPE** de um objeto AUTHINFO especifica o tipo do objeto Os objetos AUTHINFO que são usados para autenticação de conexão podem ser um dos dois tipos a seguir:

IDPWOS
O gerenciador de filas usa o sistema operacional local para autenticar o ID do usuário e a senha fornecidos por um aplicativo de conexão.

Em IBM MQ 9.3.4, esse tipo de objeto AUTHINFO também permite que um gerenciador de filas que é executado no AIX ou Linux valide tokens de autenticação. Além do objeto AUTHINFO que é usado para configurar a autenticação de conexão, o gerenciador de fila deve ser configurado para aceitar tokens de autenticação com a sub-rotina **AuthInfo** do arquivo `qm.ini` .. Para obter mais informações sobre a configuração de um gerenciador de filas para aceitar tokens de autenticação, consulte “Configurando um gerenciador de filas para aceitar tokens de autenticação usando um keystore local” na página 337

IDPWLDAP
O gerenciador de filas usa um servidor LDAP para autenticar o ID do usuário e a senha fornecidos por um aplicativo de conexão.

Nota: Não é possível especificar qualquer outro tipo de objeto de informações sobre autenticação no atributo **CONNAUTH** do gerenciador de filas

Os objetos AUTHINFO do tipo `IDPWOS` e `IDPWLDAP` são semelhantes em vários de seus atributos Os atributos descritos aqui são comuns a ambos os tipos de objetos

Os comandos MQSC de exemplo a seguir ativam a autenticação de conexão com as operações a seguir:

1. Defina um objeto AUTHINFO denominado `USE.PW`..
2. Altere o atributo **CONNAUTH** do gerenciador de filas para referir-se a esse objeto AUTHINFO
3. Emita o comando **REFRESH SECURITY** para atualizar a configuração de autenticação de conexão do gerenciador de filas O comando **REFRESH SECURITY** deve ser emitido antes que o gerenciador de filas reconheça quaisquer mudanças na configuração de autenticação de conexão

```
DEFINE AUTHINFO(USE.PW) +
AUTHTYPE(IDPWOS) +
FAILDLAY(10) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED)

ALTER QMGR CONNAUTH(USE.PW)

REFRESH SECURITY TYPE(CONNAUTH)
```

Para controlar se as credenciais são verificadas para conexões que são feitas por aplicativos ligados localmente, use o atributo AUTHINFO **CHCKLOCL** (verificar conexões locais). Para controlar se as credenciais são verificadas para conexões que são feitas por aplicativos clientes, use o atributo AUTHINFO **CHCKCLNT** (verificar conexões do cliente).

CHCKLOCL aceita os valores de `NONE` e `OPTIONALe` **CHCKCLNT** permite que o valor de `NONE` para os requisitos de autenticação seja configurado:

NONE
As credenciais de autenticação fornecidas pelos aplicativos não são verificadas.

OPCIONAL
Assegura que quaisquer credenciais fornecidas por um aplicativo sejam válidas. No entanto, não é obrigatório para os aplicativos fornecerem credenciais de autenticação. Esta opção pode ser útil durante a migração, por exemplo.

Se você:

- Forneça o nome do usuário e a senha, pois eles são autenticadas
- Não forneça o nome do usuário e a senha, a conexão é permitida
- Forneça o nome do usuário, mas não a senha que você recebe um erro.

Importante: `OPTIONAL` é o valor mínimo que pode ser configurado se você também desejar configurar uma opção mais restritiva nas regras de autenticação de canal (CHLAUTH)

Se você selecionar NONE e a conexão do cliente corresponder a um registro CHLAUTH com **CHCKCLNT** configurado como REQUIRED (ou REQDADM em plataformas diferentes de z/OS), a conexão falhará Você recebe a mensagem AMQ9793 em Multiplataformas e a mensagem CSQX793E em z/OS

Para obter mais informações sobre como usar regras de autenticação de canal para configurar opções **CHCKCLNT** mais restritivas para algumas conexões do cliente, consulte "Granularidade de configuração" na página 77.

REQUIRED
Requer que todos os aplicativos forneçam credenciais válidas.. Consulte também a nota a seguir.

REQDADM
Usuários privilegiados devem fornecer credenciais válidas, mas usuários não privilegiados são tratados como com a configuração OPTIONAL . Consulte também a nota a seguir. z/OS (Essa configuração não é permitida em sistemas z/OS.)

Nota:

Configurar **CHCKLOCL** como REQUIRED ou REQDADM significa que não é possível administrar localmente o gerenciador de filas usando o **runmqsc** (erro AMQ8135: Não autorizado), a menos que o usuário especifique o parâmetro **-u** para especificar o ID do usuário no comando **runmqsc** Com esse conjunto de parâmetros, **runmqsc** solicita a senha do usuário no console.

Da mesma forma, um usuário que executa IBM MQ Explorer no sistema local verá o erro AMQ4036 ao tentar conectar-se ao gerenciador de fila Para especificar um ID de usuário e senha, clique com o botão direito no objeto do gerenciador de filas locais e selecione **Detalhes da Conexão** > **Propriedades ...** a partir do menu. Na seção **ID do usuário** , insira o ID do usuário e a senha a serem usados, em seguida, clique em **OK**.

Considerações semelhantes se aplicam às conexões remotas com **CHCKCLNT**.

O atributo **CONNAUTH** do gerenciador de filas está em branco para os gerenciadores de filas que são migrados de versões anteriores a IBM MQ 8.0, mas é configurado como *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* para gerenciadores de filas recém-criados.. Essa definição padrão de **AUTHINFO** tem **CHCKCLNT** configurado como REQDADM por padrão

Portanto, quaisquer clientes existentes que usam um ID do usuário privilegiado para se conectar devem fornecer credenciais válidas.

Aviso: As credenciais em uma estrutura MQCSP para um aplicativo cliente são, às vezes, enviadas pela rede em texto simples. Para assegurar que as credenciais do cliente sejam protegidas, consulte "Proteção de senha do MQCSP" na página 32

Granularidade de configuração

Os atributos **CHCKLOCL** e **CHCKCLNT** do objeto AUTHINFO definem requisitos de autenticação para todas as conexões com o gerenciador de filas. Além desses atributos, as regras **CHCKCLNT** attribute on channel authentication (CHLAUTH) permitem que requisitos de autenticação mais rigorosos sejam configurados para conexões específicas do cliente que correspondem à regra CHLAUTH.

É possível configurar o valor geral **CHCKCLNT** como OPTIONAL, por exemplo, no objeto AUTHINFO e, em seguida, fazer upgrade dele para ser mais rigoroso para determinados canais, configurando **CHCKCLNT** como REQUIRED ou REQDADM na regra CHLAUTH. Por padrão, as regras CHLAUTH são definidas com **CHCKCLNT(ASQMGR)**, portanto, essa granularidade não precisa ser usada Por exemplo, esses comandos MQSC definem uma regra CHLAUTH que substitui o atributo **CHCKCLNT** do objeto AUTHINFO e uma regra CHLAUTH que não:

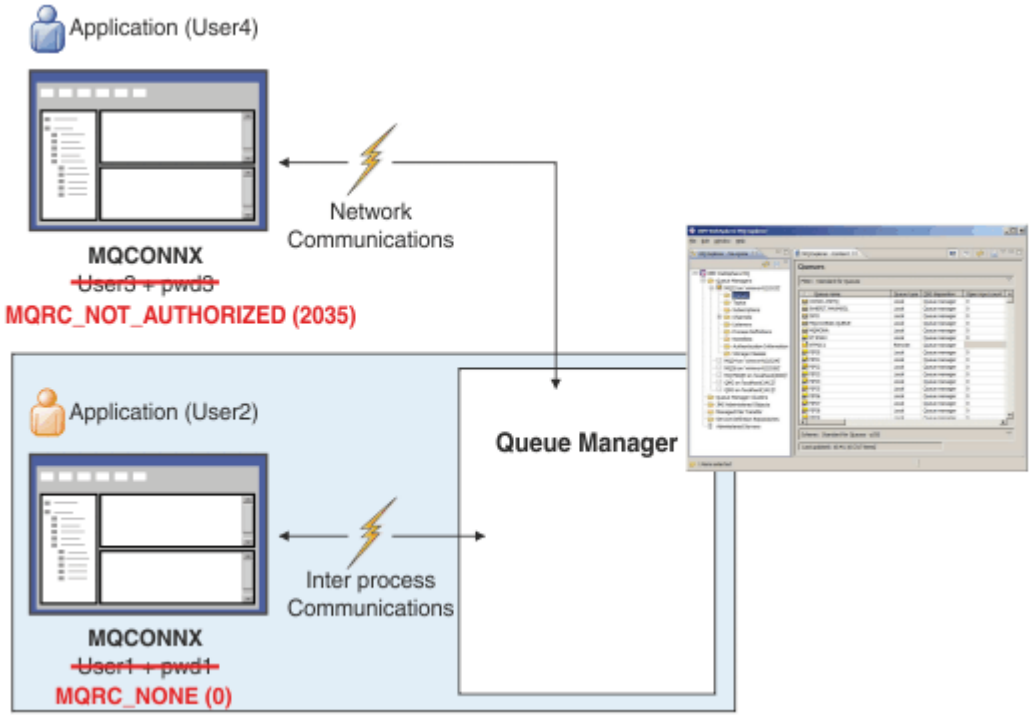
```
DEFINE AUTHINFO(USE.PW) AUTHTYPE(xxxxxx) +
CHCKCLNT(OPTIONAL)

SET CHLAUTH('*') TYPE(ADDRESSMAP) +
ADDRESS('*') USERSRC(CHANNEL) +
CHCKCLNT(REQUIRED)
```

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

Para obter mais informações sobre regras CHLAUTH, consulte “Registros de Autenticação de Canal” na página 53.

Notificação de Erro



Um erro é registrado nas seguintes situações:

- Um aplicativo não fornece credenciais de autenticação quando são necessárias.
- Um aplicativo fornece credenciais de autenticação inválida. Essa situação é tratada como um erro mesmo se a configuração indicar que é opcional para os aplicativos fornecerem credenciais.

Nota: Quando **CHCKLOCL** ou **CHCKCLNT** for configurado como NONE, as credenciais inválidas que são fornecidas pelos aplicativos não serão detectados

Autenticações com falha são mantidas pelo número de segundos especificado pelo atributo **FAILDLAY** antes de o erro ser retornado ao aplicativo. Esse atraso fornece alguma proteção de um aplicativo repetidamente tentando conectar.

O erro é registrado de várias formas:

Aplicativo
Um código de razão MQRC_NOT_AUTHORIZED (2035) é retornado ao aplicativo.

Administrator
Um administrador IBM MQ vê o evento relatado no log de erros. A mensagem de erro mostra que a conexão foi rejeitada porque as credenciais são inválidas e não porque, por exemplo, o usuário não possui autoridade de conexão.

Ferramenta de monitoração
Uma ferramenta de monitoramento também pode ser notificada da falha, se você ativar eventos de autoridade, por uma mensagem de evento na fila SYSTEM.ADMIN.QMGR.EVENT . Para ativar eventos de autoridade, emita o comando MQSC a seguir:

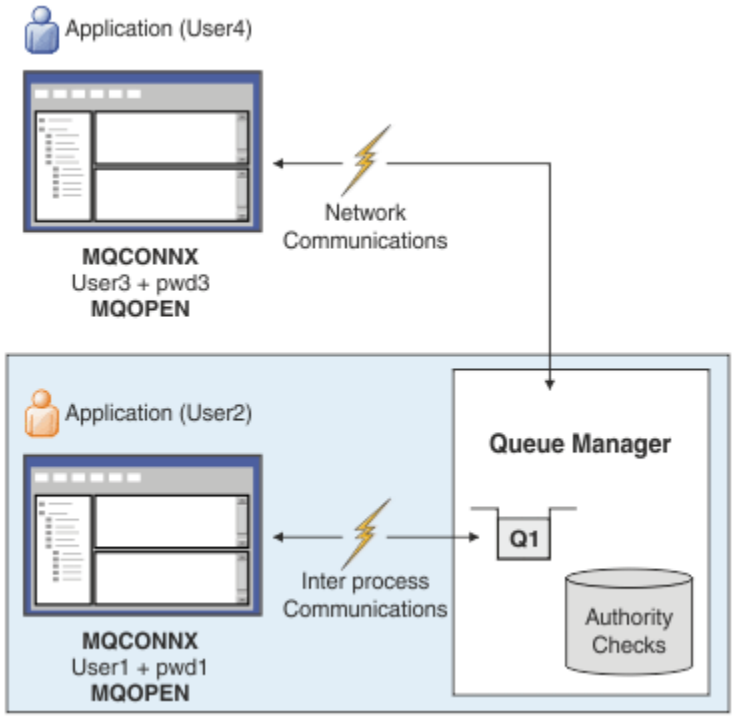
```
ALTER QMGR AUTHOREV(ENABLED)
```

Esse evento "Não autorizado" é um evento de conexão Tipo 1 e fornece os mesmos campos que outros eventos Tipo 1, com um campo extra, o ID do usuário MQCSP que foi fornecido. Se o aplicativo tiver fornecido uma senha, ela não será incluída na mensagem do evento Isso significa que há dois IDs de usuário na mensagem do evento:

- O ID do usuário no qual o aplicativo está em execução.
- O ID do usuário nas credenciais apresentadas pelo aplicativo.

Para obter mais informações sobre essa mensagem do evento. Consulte Não autorizado (tipo 1).

Adotando usuários para autorização



É possível configurar o gerenciador de filas para adotar as credenciais apresentadas pelo aplicativo como o contexto para a conexão. Adotar as credenciais significa que o ID do usuário fornecido nas credenciais de autenticação é usado para verificações de autorização, mostrado em exibições administrativas e aparece em mensagens. O atributo **ADOPTCTX** no objeto AUTHINFO controla se as credenciais são adotados como o contexto para o aplicativo Por exemplo, os comandos MQSC a seguir definem um objeto AUTHINFO denominado USE . PWD que é usado para autenticação de conexão e configuram o atributo **ADOPTCTX** como YES:

```
DEFINE AUTHINFO(USE.PWD) +
AUTHTYPE(xxxxxx) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED) +
ADOPTCTX(YES)

ALTER QMGR CONNAUTH(USE.PWD)
```

Os seguintes valores podem ser especificados para o atributo **ADOPTCTX** :

ADOPTCTX(YES)
As credenciais fornecidas pelo aplicativo são adotadas como o contexto de aplicativos para a duração da conexão. Todas as verificações de autorização para um aplicativo são feitas com o ID do usuário nas credenciais autenticadas.

Atenção: Ao usar o **ADOPTCTX(YES)** e IDs do usuário do sistema operacional local, deve-se assegurar que o ID do usuário que está sendo adotado atenda aos requisitos para IDs do

usuário no IBM MQ. Para obter informações adicionais, consulte “IDs de Usuário” na página 93.

ADOPTCTX(NO)
As credenciais fornecidas por um aplicativo são usadas apenas para autenticação no tempo de conexão. O ID do usuário sob o qual o aplicativo está em execução continua a ser usado para futuras verificações de autorização Você pode achar essa opção útil ao migrar ou, se você planeja usar outros mecanismos, como registros de autenticação de canal, para designar o identificador de usuário do canal de mensagens (MCAUSER).

Interação com Autenticação de Canal

As regras de autenticação de canal podem ser usadas para alterar o ID do usuário usado como o contexto para uma conexão de aplicativo, com base no ID do usuário recebido do cliente. Para obter um exemplo de como usar uma regra de autenticação de canal para mudar o ID do usuário associado a uma conexão, consulte “Mapeando um ID de usuário cliente para um ID do usuário MCAUSER” na página 393...

A ordem na qual as regras de autenticação de conexão e autenticação de canal são processadas é um fator significativo para determinar o contexto de segurança para conexões de aplicativo cliente do IBM MQ. O parâmetro **ChlauthEarlyAdopt** na sub-rotina **channels** do arquivo `qm.ini` controla a ordem na qual o gerenciador de filas adota o contexto de credenciais que são fornecidas pelo aplicativo e aplica regras de autenticação de canal. Para obter mais informações sobre **ChlauthEarlyAdopt**, veja Atributos da sub-rotina de canais.

Atenção: Ao usar o parâmetro **ADOPTCTX(YES)** no objeto de informações sobre autenticação, o contexto que é adotado a partir das credenciais que são fornecidas pelo aplicativo pode ser alterado pelas regras de autenticação de canal apenas se o parâmetro **ChlauthEarlyAdopt** for configurado como Y.

Para obter mais informações sobre a interação de autenticação de conexão e autenticação de canal e a ordem na qual as verificações ocorrem quando um aplicativo cliente se conecta a um gerenciador de filas, consulte “Interação de CHLAUTH e CONNAUTH” na página 59.

Conceitos relacionados
“Autenticação de conexão” na página 74
A autenticação de conexão permite que os aplicativos forneçam as credenciais de autenticação quando se conectam a um gerenciador de filas O gerenciador de filas valida as credenciais O ID do usuário fornecido nas credenciais também pode ser adotado para uso em verificações de autorização para os recursos que o aplicativo acessa..

“Autenticação de conexão: Mudanças no aplicativo” na página 80

“Autenticação de conexão: Repositórios do usuário” na página 81
Para cada um de seus gerenciadores de filas, é possível escolher diferentes tipos de objeto de informações sobre autenticação para autenticar os IDs do usuário e senhas.

Autenticação de conexão: Mudanças no aplicativo

Um aplicativo que usa a interface da fila de mensagens (MQI) pode fornecer um ID do usuário e uma senha na estrutura dos parâmetros de segurança de conexão (MQCSP) quando MQCONNX é chamado. Em outras interfaces de programação de aplicativos, a estrutura MQCSP é geralmente construída em nome do aplicativo pelas bibliotecas do IBM MQ

V 9.4.0 No IBM MQ 9.3.4, os aplicativos clientes que se conectam a um gerenciador de filas que é executado em sistemas AIX ou Linux também podem enviar um token de autenticação na estrutura MQCSP como um meio alternativo de identificação.

O ID do usuário e a senha, ou token de autenticação, são passados para verificação para o gerenciador de autoridade de objetos (OAM) fornecido com o gerenciador de filas ou o componente de serviço de autorização fornecido com o gerenciador de filas em sistemas z/OS . Você não tem que gravar sua própria interface customizada.

Se o aplicativo estiver em execução como um cliente, o ID do usuário e a senhaou token de autenticação, também será transmitido para as saídas de segurança do lado do cliente e do lado do servidor para processamento. Eles também podem ser usados para configurar o atributo message channel agent user identifier (MCAUSER) de uma instância de canal.

Aviso: As credenciais em uma estrutura MQCSP para um aplicativo cliente são, às vezes, enviadas pela rede em texto simples. Para assegurar que as credenciais do aplicativo cliente sejam protegidas, consulte “Proteção de senha do MQCSP” na página 32

Usando a sequência XAOPEN para fornecer um ID do usuário e senha, é possível evitar ter que mudar o código do aplicativo.

Nota:

Em IBM WebSphere MQ 6.0, a saída de segurança permite que o MQCSP seja configurado. Portanto, os clientes nesse nível ou mais recente não precisam ser atualizados.

No entanto, em versões do IBM MQ anteriores à IBM MQ 8.0, o MQCSP não colocou restrições sobre o ID do usuário e a senha que foram fornecidos pelo aplicativo. Ao utilizar esses valores com recursos fornecidos pelo IBM MQ, há limites que se aplicam ao uso desses recursos, mas se você estiver apenas passando-os para as suas próprias saídas, esses limites não se aplicam.

Conceitos relacionados

“Autenticação de conexão” na página 74
A autenticação de conexão permite que os aplicativos forneçam as credenciais de autenticação quando se conectam a um gerenciador de filas O gerenciador de filas valida as credenciais O ID do usuário fornecido nas credenciais também pode ser adotado para uso em verificações de autorização para os recursos que o aplicativo acessa..

“Autenticação de conexão: configuração” na página 75
Um gerenciador de filas pode ser configurado para autenticar credenciais que são fornecidas por um aplicativo quando ele se conecta..

“Autenticação de conexão: Repositórios do usuário” na página 81
Para cada um de seus gerenciadores de filas, é possível escolher diferentes tipos de objeto de informações sobre autenticação para autenticar os IDs do usuário e senhas.

Autenticação de conexão: Repositórios do usuário

Para cada um de seus gerenciadores de filas, é possível escolher diferentes tipos de objeto de informações sobre autenticação para autenticar os IDs do usuário e senhas.

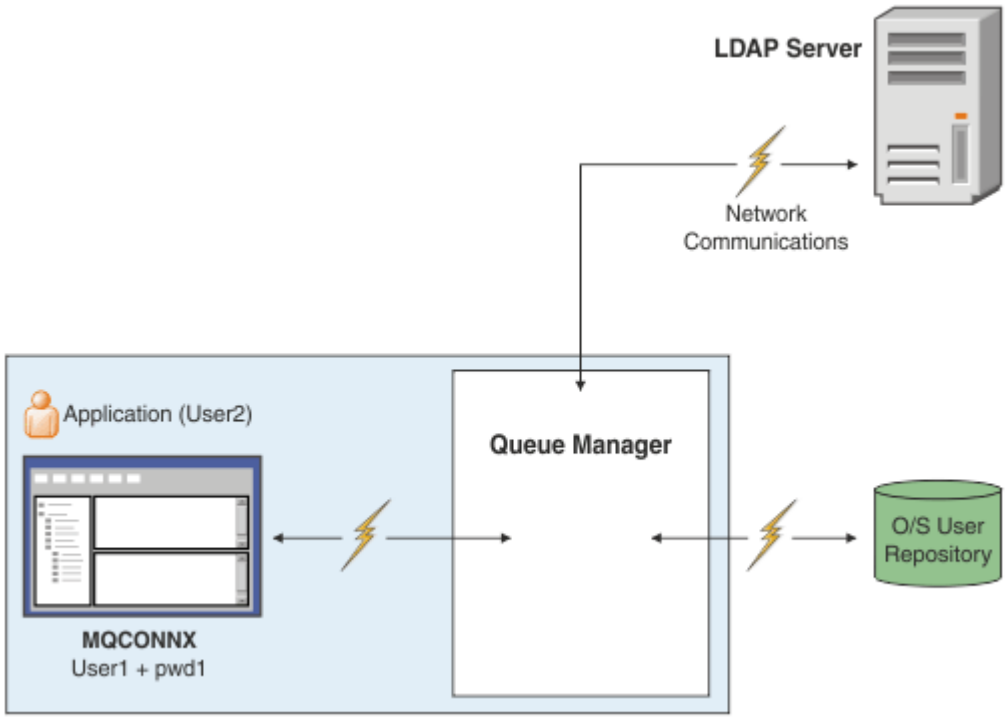


Figura 7. Tipos de objetos de informações de autenticação

```
DEFINE AUTHINFO(USE.OS) AUTHTYPE(IDPWOS)
DEFINE AUTHINFO(USE.LDAP) +
AUTHTYPE(IDPWLDAP) +
CONNAME('ldap1(389),ldap2(389)') +
LDAPUSER('CN=QMGR1') +
LDAPPWD('passw0rd') SECCOMM(YES)
```

Há dois tipos de objeto de informações sobre autenticação, conforme representado no diagrama:

- `IDPWOS` é usado para indicar que o gerenciador de filas usa o sistema operacional local para autenticar o ID do usuário e a senha. Se você optar por usar o sistema operacional local, será necessário configurar os atributos comuns, conforme descrito nos tópicos anteriores.
- `IDPWLDAP` é usado para indicar que o gerenciador de filas usa um servidor LDAP para autenticar o ID do usuário e a senha. Se você optar por usar um servidor LDAP, mais informações serão fornecidas nesse tópico.

Somente um tipo de objeto de informações sobre autenticação pode ser escolhido para cada gerenciador de filas usar nomeando o objeto apropriado no atributo **CONNAUTH** do gerenciador de filas.

Usando um servidor LDAP para autenticação.

Configure o campo **CONNAME** para o endereço do servidor LDAP para o gerenciador de filas. É possível fornecer mais endereços para o servidor LDAP em uma lista separada por vírgula, que pode ajudar com redundância se o servidor LDAP não fornecer esse recurso em si.

Configure o servidor LDAP necessários ID e senha nos campos **LDAPUSER** e **LDAPPWD** para que o gerenciador de filas possa acessar o servidor LDAP e procurar informações sobre registros do usuário.

Conexão segura para um servidor LDAP

Diferente dos canais, não há parâmetro **SSLCIPH** para ativar o uso do TLS para a comunicação com o servidor LDAP. Neste caso do IBM MQ está atuando como um cliente para o servidor LDAP, portanto muito da configuração é feito no servidor LDAP. Alguns parâmetros existentes no IBM MQ são usados para configurar como essa conexão funciona.

Configure o campo **SECCOMM** para controlar se a conectividade com o servidor LDAP usa o TLS.

Além desse atributo, os atributos **SSLFIPS** e **SUITEB** do gerenciador de filas restringem o conjunto de especificações de criptografia que são escolhidas. O certificado que é usado para identificar o gerenciador de filas para o servidor LDAP é o certificado do gerenciador de filas `ibmwebspheremq` *`qmgr-name`* ou o valor do atributo **CERTLABL**. Consulte Rótulos de certificado digital para obter detalhes.

Repositório de Usuário LDAP

Ao usar um repositório do usuário LDAP, existe mais alguma configuração a ser feita no gerenciador de filas diferente de apenas informar a ele onde localizar o servidor LDAP.

IDs de usuário definidos em um servidor LDAP têm uma estrutura hierárquica que os identificam exclusivamente. Portanto, um aplicativo pode se conectar ao gerenciador de filas e apresentar seu ID do usuário como o ID do usuário hierárquico completo.

Entretanto, para simplificar as informações que um aplicativo deve fornecer, é possível configurar o gerenciador de filas para presumir que a primeira parte da hierarquia seja comum a todos os IDs e para incluir automaticamente isso antes do ID reduzido fornecido pelo aplicativo. O gerenciador de filas pode, então, apresentar um ID completo ao servidor LDAP.

Configure BASEDNU como o ponto inicial que a procura do LDAP faz para o ID na hierarquia do LDAP. Ao configurar BASEDNU, deve-se assegurar de que somente um resultado será retornado quando procurar o ID na hierarquia do LDAP.

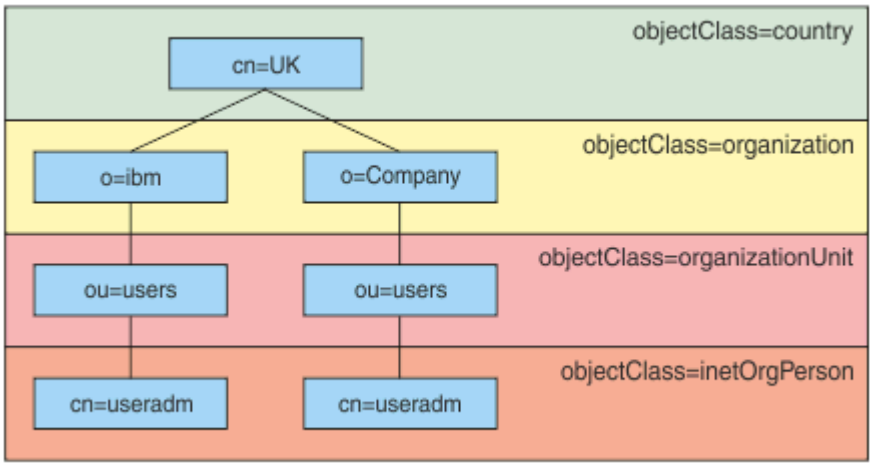


Figura 8. Uma hierarquia LDAP de exemplo

Por exemplo, em Figura 8 na página 83, BASEDNU pode ser configurado como "ou=users,o=ibm,c=UK" ou ",o=ibm,c=UK". Entretanto, como um nome distinto que contém "cn=useradm" existe em ambas as ramificações "o=ibm" e "o=Company", BASEDNU não pode ser configurado como "c=UK". Por motivos de desempenho e segurança, use o ponto mais alto em sua hierarquia de LDAP a partir do qual seja possível referenciar todos os IDs do usuário necessários. Neste exemplo, isso é "ou=users,o=ibm,c=UK".

Seu aplicativo pode enviar ao gerenciador de filas o ID do usuário sem fornecer o nome do atributo LDAP, CN=, por exemplo. Se você configurar USRFIELD para o nome do atributo LDAP, esse valor é incluído como um prefixo para o ID do usuário que vem do aplicativo. Isso pode ser um auxílio de migração útil ao mover dos IDs do usuário do sistema operacional para IDs do usuário LDAP, pois o aplicativo pode apresentar a mesma sequência em ambos os casos e é possível evitar mudar o aplicativo.

Portanto, o ID do usuário completo apresentado ao servidor LDAP é semelhante a:

```
USRFIELD = ID_from_application BASEDNU
```

Conceitos relacionados

“Autenticação de conexão” na página 74
A autenticação de conexão permite que os aplicativos forneçam as credenciais de autenticação quando se conectam a um gerenciador de filas O gerenciador de filas valida as credenciais O ID do usuário fornecido nas credenciais também pode ser adotado para uso em verificações de autorização para os recursos que o aplicativo acessa..

“Autenticação de conexão: configuração” na página 75
Um gerenciador de filas pode ser configurado para autenticar credenciais que são fornecidas por um aplicativo quando ele se conecta..

“Autenticação de conexão: Mudanças no aplicativo” na página 80

Saída de segurança do lado do cliente para inserir o ID do usuário e a senha (`mqccred`)

Se você tiver quaisquer aplicativos cliente que são necessários para enviar um ID do usuário ou senha, mas ainda não é possível mudar a origem, há uma saída de segurança fornecida com o IBM MQ 8.0 denominada **`mqccred`** que é possível usar. **`mqccred`** fornece um ID do usuário e senha em nome do aplicativo cliente, a partir de um arquivo `.ini`. Esse ID do usuário e a senha são enviados ao gerenciador de filas que, se configurado para fazer isso, os autenticará.

Visão Geral

`mqccred` é uma saída de segurança que é executada na mesma máquina que seu aplicativo cliente. Ela permite que informações de ID do usuário e senha sejam fornecidas em nome do aplicativo cliente, em que as informações não estão sendo fornecidas pelo próprio aplicativo. As informações do ID do usuário e senha são fornecidas em uma estrutura conhecida como Connection Security Parameters (MQCSP) e serão autenticados pelo gerenciador de filas, se a conexão de autenticação for configurada.

As informações do ID do usuário e senha são recuperadas de um arquivo `.ini` na máquina cliente. As senhas no arquivo são protegidas por ofuscação usando o comando **`runmqccred`** e também assegurando que as permissões de arquivo no arquivo `.ini` sejam configuradas de tal forma que somente o ID do usuário executando o aplicativo cliente (e, portanto, a saída) seja capaz de lê-las.

Local

`mqccred` é instalado:

Plataformas Windows
No diretório *`installation_directory`*`\Tools\c\Samples\mqccred\`

Plataformas AIX and Linux
No diretório *`installation_directory`*`/samp/mqccred`

Notas: A saída:

1. Age apenas como uma saída de segurança do canal e precisa ser a única tal saída definida em um canal.
2. É geralmente chamada por meio do Client Channel Definition Table (CCDT), mas um cliente Java pode ter a saída mencionada diretamente nos objetos JNDI ou a saída pode ser configurada para aplicativos que constroem manualmente a estrutura MQCD.
3. Deve-se copiar os programas **`mqccred`** e **`mqccred_r`** para o diretório `var/mqm/exits`.

   Por exemplo, em um sistema AIX ou Linux de 64 bits, emita o comando:

   ```
   cp installation_directory/samp/mqccred/lib64/* /var/mqm/exits
   ```

   Consulte Um exemplo passo a passo de como testar mqccred para obter mais informações.
4. É capaz de executar em versões anteriores do IBM MQ, desde o IBM WebSphere MQ 7.0.1.

Configurando IDs do Usuário e Senhas

O arquivo `.ini` contém sub-rotinas para cada gerenciador de filas, com uma configuração global para gerenciadores de filas não especificados. Cada sub-rotina contém o nome do gerenciador de filas, um ID do usuário e um texto simples ou uma senha ofuscada.

Deve-se editar o arquivo `.ini` manualmente, usando o editor de sua escolha e incluir o atributo de senha de texto simples para as sub-rotinas. Execute o programa **runmqccred** fornecido, que usa o arquivo `.ini` e substitui o atributo **Password** com o atributo **OPW**, uma forma ofuscada da senha.

Consulte runmqccred para uma descrição do comando e seus parâmetros.

O arquivo `mqccred.ini` contém seu ID do usuário e senha.

Um arquivo de modelo `.ini` é fornecido no mesmo diretório que a saída para fornecer um ponto de início para sua empresa.

Por padrão, este arquivo será procurado em `$HOME/.mqs/mqccred.ini`. Se você gostaria de localizá-lo em outro lugar, é possível usar a variável de ambiente *MQCCRED* para apontar para ele:

```
MQCCRED=C:\mydir\mqccred.ini
```

Se você usar *MQCCRED*, a variável deve incluir o nome completo do arquivo de configuração, incluindo qualquer arquivo `.ini`. Como esse arquivo contém senhas (mesmo se ofuscadas), é esperado que você proteja o arquivo usando privilégios do sistema operacional para assegurar que as pessoas não autorizadas não possam lê-la. Se você não tem a permissão de arquivo correta, a saída não executará com êxito.

Se o aplicativo já tiver fornecido uma estrutura MQCSP a saída normalmente respeita isso e não irá inserir quaisquer informações a partir do arquivo `.ini`. Entretanto, você pode substituir esse usando o atributo **Force** na sub-rotina.

Configurar **Force** para o valor *TRUE* remove o ID do usuário e senha fornecidos pelo aplicativo e substitui aqueles com a versão do arquivo ini.

Também é possível configurar o atributo **Force** na seção global do arquivo para configurar o valor padrão desse arquivo.

O valor padrão para **Force** é *FALSE*.

Você pode fornecer um ID do usuário e uma senha para todos os gerenciadores de filas, ou para cada gerenciador de filas individual. Este é um exemplo de um arquivo `mqccred.ini`:

```
# comments are permitted
AllQueueManagers:
User=abc
OPW=%^&aervrgtsr

QueueManager:
Name=QMA
User=user1
OPW=H&^dbgfh

Force=TRUE

QueueManager:
Name=QMB
User=user2
password=passw0rd
```

Notas:

1. As definições de gerenciador de filas individual têm precedência sobre a configuração global.
2. Os atributos fazem distinção entre maiúsculas e minúsculas.

Restrições

Quando essa saída estiver em uso, o ID do usuário local da pessoa que está executando o aplicativo não fluirá do cliente para o servidor. As únicas informações de identidade disponíveis são a partir do conteúdo do arquivo ini.

Portanto, deve-se configurar o gerenciador de filas para usar **ADOPTCTX(YES)** ou mapear a solicitação de conexão de entrada para um ID do usuário apropriado por meio de um dos mecanismos disponíveis, por exemplo, "Registros de Autenticação de Canal" na página 53.

Importante: Se você incluir novas senhas ou atualizar antigas, o comando **runmqccred** somente processará quaisquer senhas de texto simples, deixando as ofuscadas intocadas.

Depurando

A saída grava para o rastreio do IBM MQ padrão quando ele está ativado.

Para ajudar na depuração de problemas de configuração, a saída também pode gravar diretamente para stdout.

Nenhuma configuração de dados da saída de segurança do canal (**SCYDATA**) é normalmente requerida para o canal. Entretanto, é possível especificar:

ERRO
Imprime somente informações sobre condições de erro, como não ser capaz de localizar o arquivo de configuração.

DEBUG
Exibe estas condições de erro e algumas trilhas de auditoria adicionais.

NOCHECKS
Efetua bypass das restrições sobre as permissões de arquivo e a restrição adicional de que o arquivo `.ini` não deve conter quaisquer senhas desprotegidas.

É possível colocar um ou mais desses elementos no campo **SCYDATA**, separados por vírgulas, em qualquer ordem. Por exemplo, `SCYDATA=(NOCHECKS,DEBUG)`.

Observe que os itens fazem distinção entre maiúsculas e minúsculas e devem ser digitados em letras maiúsculas.

Usando o `mqccred`

Assim que tiver o seu arquivo configurado, é possível chamar a saída de canal atualizando a sua definição de canal de conexão do cliente para incluir o atributo `SCYEXIT('mqccred(ChlExit)')`:

```
DEFINE CHANNEL(channelname) CHLTYPE(clntconn) +
CONNAME(remote machine) +
QMNAME(remote qmgr) +
SCYEXIT('mqccred(ChlExit)') +
REPLACE
```

Referências relacionadas
SCYDATA
SCYEXIT
runmqccred

Autenticação de conexão com o cliente Java

A autenticação de conexão é um recurso no IBM MQ que permite configurar gerenciadores de filas de modo que eles possam autenticar aplicativos usando um ID de usuário e uma senha fornecidos. Quando o aplicativo é um aplicativo Java que está usando o transporte do cliente, a autenticação de conexão pode ser executada no modo de compatibilidade ou no modo de autenticação MQCSP.

O ID do usuário e a senha a serem autenticados são especificados pelo aplicativo usando um dos métodos a seguir:

- Em um aplicativo IBM MQ classes for Java, na classe `MQEnvironment` ou nas propriedades Hashtable passadas para o construtor `com.ibm.mq.MQQueueManager`.
- Em um aplicativo IBM MQ classes for JMS, como argumentos para o método `createConnection(String username, String Password)` ou `createContext(String username, String password)`.

Modo de autenticação MQCSP

Nesse modo, o ID do usuário do lado do cliente sob o qual o aplicativo é executado é enviado para o gerenciador de filas, bem como o ID do usuário e a senha a serem autenticados. O IBM MQ classes for Java e o IBM MQ classes for JMS enviam o ID do usuário e a senha a serem autenticados para o gerenciador de filas em uma estrutura MQCSP.

O ID do usuário e a senha estão disponíveis para uma saída de segurança de conexão do servidor dentro da estrutura MQCSP. O endereço da estrutura MQCSP pode ser encontrado no campo **SecurityParms** da estrutura MQCXP do canal.

O modo de autenticação MQCSP tem os benefícios a seguir:

- O comprimento máximo do ID do usuário a ser autenticado é de 1.024 caracteres.
- O comprimento máximo da senha para autenticação é de 256 caracteres.
- As verificações de autorização de acesso ao uso de recursos do IBM MQ podem ser executadas usando o ID do usuário do lado do cliente sob o qual o aplicativo é executado quando o objeto de informações de autenticação usado para controlar a autenticação de conexão no gerenciador de filas está configurado com ADOPTCTX(NO).

Compatibility Mode

Antes do IBM MQ 8.0, o cliente Java podia enviar um ID do usuário e senha por meio do canal de conexão do cliente para o canal de conexão do servidor e fornecer a eles uma saída de segurança nos campos **RemoteUserIdentifier** e **RemotePassword** da estrutura MQCD. No modo de compatibilidade, esse comportamento é retido.

Você pode usar este modo em combinação com a autenticação de conexão e migrar para fora de quaisquer saídas de segurança que foram anteriormente usadas para executar a mesma tarefa.

Esse modo tem as restrições a seguir:

- O comprimento do ID do usuário e da senha deve ser de 12 caracteres ou menos. IDs de usuário com mais de 12 caracteres de comprimento são truncados para 12 caracteres. Isso pode fazer com que a conexão falhe com o código de razão MQRC_NOT_AUTHORIZED.
- O ID do usuário do lado do cliente sob o qual o aplicativo é executado não é enviado ao gerenciador de filas. Deve-se definir ADOPTCTX(YES) no objeto de informações sobre autenticação que é usado para controlar a autenticação de conexão no gerenciador de filas ou usar outro método, como uma regra de autenticação de canal baseada em um certificado TLS para configurar o ID do usuário MCA do canal que está marcado para autorização de uso de recursos IBM MQ.

Modo de autenticação padrão

O modo de autenticação padrão usado por um aplicativo cliente IBM MQ classes for Java ou IBM MQ classes for JMS varia dependendo se o aplicativo especifica um ID do usuário e uma senha.

- Se um ID do usuário e senha forem especificados, a autenticação MQCSP será usada por padrão.
- Se um ID do usuário for especificado, mas não uma senha, o modo de compatibilidade será usado por padrão.
- Se nenhum ID do usuário for especificado, o modo de compatibilidade será sempre usado.

Nos casos em que um ID do usuário é especificado, um modo de autenticação específico pode ser escolhido pelo aplicativo para cada conexão individual ou configurado globalmente antes de o aplicativo ser iniciado, conforme descrito em “Escolhendo o modo de autenticação” na página 88.

Nota: Os aplicativos que usam o IBM MQ classes for JMS podem ser afetados pela mudança para o modo de autenticação padrão no IBM MQ 9.3.0. Após fazer upgrade do IBM MQ classes for JMS para o IBM MQ 9.3.0, que usavam anteriormente o modo de compatibilidade por padrão usarão a autenticação MQCSP em vez disso. Isso pode fazer com que os aplicativos que se conectaram anteriormente com êxito a um gerenciador de filas falhem ao se conectar com um `JMSException` contendo o código de razão 2035 (MQRC_NOT_AUTHORIZED). Se isso ocorrer, use um dos métodos descritos em "Escolhendo o modo de autenticação" na página 88 para especificar que o aplicativo usa o modo de compatibilidade.

Os aplicativos Java que se conectam ao gerenciador de filas usando ligações locais sempre usam o modo de autenticação MQCSP.

Escolhendo o modo de autenticação

O modo de autenticação usado por aplicativos clientes Java que especificam um ID do usuário ao se conectar ao gerenciador de filas pode ser especificado usando um dos métodos a seguir. Esses métodos são listados na ordem decrescente de precedência. Se o modo de autenticação não for especificado usando qualquer um desses métodos, então o modo de autenticação padrão será usado.

Nota: O uso desses métodos para selecionar o modo de autenticação foi esclarecido em IBM MQ 9.3.0 Em alguns casos, o modo de autenticação usado por um aplicativo cliente Java pode mudar quando o IBM MQ classes for Java ou o IBM MQ classes for JMS é submetido a upgrade para o IBM MQ 9.3.0. Isso pode fazer com que os aplicativos que se conectaram anteriormente com êxito a um gerenciador de filas falhem ao se conectar com um `JMSException` contendo o código de razão 2035 (MQRC_NOT_AUTHORIZED). Se isso ocorrer, use um dos métodos a seguir para selecionar o modo de autenticação necessário.

- Especifique o modo de autenticação para cada conexão individual configurando a propriedade apropriada no aplicativo antes de se conectar ao gerenciador de filas.
  - Ao usar o IBM MQ classes for Java, configure a propriedade *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* nas propriedades Hashtable passadas para o construtor `com.ibm.mq.MQQueueManager`.
  - Ao usar o IBM MQ classes for JMS, configure a propriedade *JmsConstants.USER_AUTHENTICATION_MQCSP* no connection factory apropriado antes de criar a conexão.

  Configure o valor dessas propriedades para um dos valores a seguir:

  true
  Use o modo de autenticação MQCSP ao autenticar com um gerenciador de filas.

  false
  Use o modo de compatibilidade ao autenticar com um gerenciador de filas.
- Especifique o modo de autenticação para todas as conexões do cliente feitas por um aplicativo configurando a propriedade do sistema *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java ao iniciar o aplicativo. Configure o valor da propriedade para um dos valores a seguir:

  Y
  Use o modo de autenticação MQCSP ao autenticar com um gerenciador de filas.

  N
  Use o modo de compatibilidade ao autenticar com um gerenciador de filas.

  Por exemplo, o comando a seguir configura a propriedade para selecionar o modo de compatibilidade e inicia um aplicativo Java:

  ```
  java -Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=N application_name
  ```
- Especifique o modo de autenticação para todas as conexões do cliente feitas por aplicativos iniciados no mesmo ambiente configurando a variável de ambiente *com.ibm.mq.jmqi.useMQCSPauthentication* no ambiente no qual o aplicativo é iniciado. Configure o valor da variável de ambiente para um dos valores a seguir:

Y
Use o modo de autenticação MQCSP ao autenticar com um gerenciador de filas.

N
Use o modo de compatibilidade ao autenticar com um gerenciador de filas.

- Especifique o modo de autenticação para todos os aplicativos que usam um arquivo de configuração do cliente IBM MQ MQI client específico especificando o atributo **useMQCSPauthentication** na sub-rotina JMQI do arquivo de configuração do cliente. Configure o valor do atributo para um dos valores a seguir:

  SIM
  Use o modo de autenticação MQCSP ao autenticar com um gerenciador de filas.

  NÃO
  Use o modo de compatibilidade ao autenticar com um gerenciador de filas.

  Para obter mais informações sobre o atributo **useMQCSPauthentication** , consulte Sub-rotina JMQI do arquivo de configuração do cliente.

Escolhendo o modo de autenticação no IBM MQ Explorer

Como o IBM MQ Explorer é um aplicativo Java, esses dois modos, o modo de compatibilidade e o modo de autenticação MQCSP, também são aplicáveis a ele.

O modo de autenticação do MQCSP é o padrão

Nos painéis nos quais a identificação de usuário é fornecida, há uma caixa de seleção para ativar ou desativar o modo de compatibilidade:

- Por padrão, essa caixa de seleção não é selecionada Para usar o modo de compatibilidade, marque essa caixa de seleção.

Conceitos relacionados

“Autenticação de conexão” na página 74
A autenticação de conexão permite que os aplicativos forneçam as credenciais de autenticação quando se conectam a um gerenciador de filas O gerenciador de filas valida as credenciais O ID do usuário fornecido nas credenciais também pode ser adotado para uso em verificações de autorização para os recursos que o aplicativo acessa..

“Autenticação de conexão: Mudanças no aplicativo” na página 80

“Autenticação de conexão: Repositórios do usuário” na página 81
Para cada um de seus gerenciadores de filas, é possível escolher diferentes tipos de objeto de informações sobre autenticação para autenticar os IDs do usuário e senhas.

Segurança de mensagem no IBM MQ

A segurança de mensagem na infraestrutura do IBM MQ é fornecida pelo Advanced Message Security.

Advanced Message Security (AMS) expande os serviços de segurança do IBM MQ para fornecer assinatura e criptografia de dados no nível de mensagem. Os serviços expandidos garantem que os dados da mensagem não foram modificados entre quando foram originalmente colocados em uma fila e quando foram recuperados. Além disso, o AMS verifica se um emissor de dados da mensagem está autorizado a colocar mensagens assinadas em uma fila de destino.

Conceitos relacionados

“Advanced Message Security” na página 608
O Advanced Message Security (AMS) é um componente do IBM MQ que fornece um alto nível de proteção para dados sensíveis que fluem por meio da rede do IBM MQ, ao mesmo tempo em que não impacta os aplicativos finais.

Planejando para seus requisitos de segurança

Esta coleção de tópicos explica o que é necessário considerar ao planejar a segurança em um ambiente do IBM MQ.

É possível usar o IBM MQ para uma grande variedade de aplicativos em uma gama de plataformas. Os requisitos de segurança provavelmente serão diferentes para cada aplicativo. Para alguns, a segurança será uma consideração crítica.

O IBM MQ fornece um intervalo de serviços de segurança de nível de link, incluindo suporte para Segurança da Camada de Transporte (TLS).

Deve-se considerar determinados aspectos de segurança ao planejar a instalação do IBM MQ:

- Multi Em Multiplataformas, se você ignorar esses aspectos e não fizer nada, não poderá usar o IBM MQ.
- z/OS No z/OS, o efeito de ignorar esses aspectos é que os recursos do IBM MQ ficam desprotegidos. Ou seja, todos os usuários podem acessar e mudar todos os recursos do IBM MQ .

Autoridade para administrar o IBM MQ

Os administradores do IBM MQ precisam de autoridade para:

- Emita comandos para administrar o IBM MQ
- Usar o IBM MQ Explorer
- IBM i Usar comandos e painéis administrativos do IBM i.
- z/OS Use os painéis de operações e de controle no z/OS
- z/OS Use o programa utilitário IBM MQ, CSQUTIL, em z/OS
- z/OS Acesse os conjuntos de dados do gerenciador de filas no z/OS

Para obter informações adicionais, consulte:

- ALW “Autoridade para administrar o IBM MQ no AIX, Linux, and Windows” na página 407
- IBM i “Autoridade para administrar o IBM MQ no IBM i” na página 95
- z/OS “Authority to administer IBM MQ on z/OS” na página 96

autoridade para trabalhar com objetos do IBM MQ

Os aplicativos podem acessar os seguintes objetos do IBM MQ, emitindo chamadas de MQI:

- Gerenciadores de filas
- Filas
- Processos
- Listas de Nomes
- tópicos

Os aplicativos podem também usar comandos de formato de comando programável (PCF) para acessar esses objetos do IBM MQ e para acessar os objetos de informações sobre autenticação de canais e autenticação também. Esses objetos podem ser protegidos pelo IBM MQ, de modo que os IDs de usuários associados ao aplicativo precisam de autoridade para acessá-los.

Para obter mais informações, consulte “Autorização para aplicativos usarem o IBM MQ” na página 98.

Segurança de canal

Os IDs de usuário associados a agentes do canal de mensagem (MCAs) necessitam de autoridade para acessar vários recursos do IBM MQ. Por exemplo, um MCA precisa ser capaz de conectar-se a um gerenciador de filas. Se estiver enviando MCA, deverá estar apto a abrir a fila de transmissão do canal. Se for um MCA de recepção, precisa ser capaz de abrir as filas de destino. Os IDs de usuário associados aos aplicativos que precisam administrar canais, inicializadores de canais e listeners precisam de autoridade para usar os comandos do PCF relevantes. No entanto, a maioria dos aplicativos não precisa desse acesso.

Para obter mais informações, consulte "Autorização de canal" na página 120.

Considerações Adicionais

É necessário considerar os seguintes aspectos de segurança apenas se você estiver usando certas funções do IBM MQ ou extensões do produto base:

- "Segurança para Clusters de Gerenciadores de Filas" na página 132
- "Segurança para o Publicar/assinar do IBM MQ" na página 133

Planejando a identificação e a autenticação

Decida quais IDs do usuário usar e como e em que níveis você deseja aplicar controles de autenticação.

Deve-se como você identificará os usuários dos seus aplicativos IBM MQ, tendo em conta que os sistemas operacionais suportam diferentes IDs de usuário de comprimentos diferentes. É possível usar registros de autenticação de canal para mapear de um ID do usuário para outro, ou para especificar um ID do usuário com base em alguns atributos da conexão. Os canais do IBM MQ que usam TLS usam certificados digitais como um mecanismo para identificação e autenticação. Cada certificado digital tem um nome distinto do assunto que pode ser mapeado para identidades específicas usando registros de autenticação de canal. Além disso, os certificados de autoridade de certificação no repositório de chaves determinam quais certificados digitais podem ser usados para autenticar para o IBM MQ. Para obter informações adicionais, consulte:

- "Mapeando um Gerenciador de Filas Remoto para um ID do Usuário MCAUSER" na página 392
- "Mapeando um ID de usuário cliente para um ID do usuário MCAUSER" na página 393
- "Mapeando um Nome Distinto de SSL ou TLS para um ID do Usuário MCAUSER" na página 394
- "Mapeando um Endereço IP para um ID do Usuário MCAUSER" na página 396

Planejando a Autenticação para um Aplicativo Cliente

É possível aplicar controles de autenticação em quatro níveis: no nível de comunicações, em saídas de segurança, com registros de canal de autenticação, e em termos de identificação que é transmitida para uma saída de segurança.

Há quatro níveis de segurança a serem considerados. O diagrama mostra um IBM MQ MQI client que está conectado a um servidor. A segurança é aplicada em quatro níveis, conforme descrito no texto a seguir. MCA é um Agente do Canal de Mensagem.

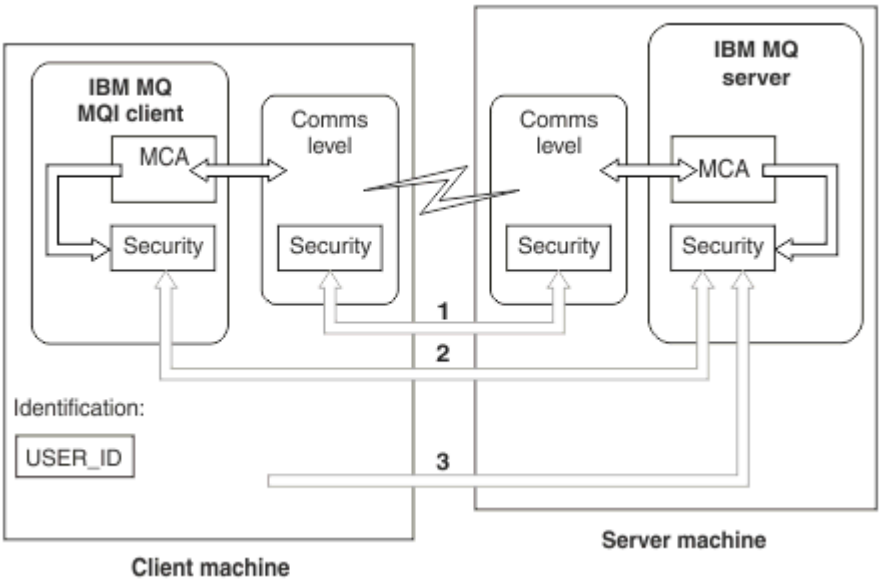


Figura 9. Segurança em uma Conexão de Cliente/Servidor

1. Nível de comunicações

   Veja a seta 1. Para implementar a segurança no nível de comunicações, use TLS. Para obter mais informações, consulte “Protocolos de segurança criptográficos: TLS” na página 18

2. Registros de Autenticação de Canal

   Ver as setas 2 e 3. A autenticação pode ser controlada com o uso do endereço IP ou nomes distintos TLS no nível de segurança. Um ID do usuário também pode ser bloqueado ou um ID do usuário declarado pode ser mapeado para um ID de usuário válido. Uma descrição completa é fornecida em “Registros de Autenticação de Canal” na página 53.

3. Autenticação de conexão

   Veja a seta 3. O cliente envia um ID do usuário e uma senhaou um símbolo de autenticação. Para obter mais informações, consulte “Autenticação de conexão: configuração” na página 75.

4. Saídas de Segurança do Canal

   Veja a seta 2. As saídas de segurança do canal para a comunicação do cliente para o servidor podem funcionar da mesma forma que para a comunicação do servidor para servidor. Um par de protocolos independentes de saída podem ser gravados para fornecer autenticação mútua entre o cliente e o servidor. Uma descrição completa é fornecida em Programas de saída de segurança de canal.

5. Identificação que é transmitida a uma saída de segurança do canal

   Veja a seta 3. Na comunicação do cliente para o servidor, as saídas de segurança do canal não têm que operar como um par. A saída no IBM MQ ao lado do cliente podem ser omitidos. Neste caso, o ID do usuário é colocado no descritor de canal (MQCD) e a saída de segurança do lado do servidor pode alterá-lo, se necessário.

   IBM MQ MQI clients também envia informações adicionais para auxiliar a identificação.

   - O ID do usuário que é transmitido ao servidor é o ID do usuário com login efetuado atualmente no cliente.
   - O ID de segurança do usuário com logon efetuado atualmente.

   Os valores do ID do usuário e, se disponível, o ID de segurança, podem ser usados por uma saída de segurança do servidor para estabelecer a identidade do IBM MQ MQI client.

No IBM MQ 8.0, é possível enviar senhas que são incluídas na estrutura do MQCSP.

Linux V 9.4.0 AIX No IBM MQ 9.3.4, IBM MQ MQI clients a conexão com IBM MQ gerenciadores de filas em execução nos sistemas AIX ou Linux também pode enviar tokens de autenticação na estrutura MQCSP.

Aviso: Em alguns casos, a senha ou token de autenticação em uma estrutura MQCSP para um aplicativo cliente é enviada pela rede em texto simples. Para assegurar que as senhas do aplicativo cliente e os tokens de autenticação sejam protegidos adequadamente, consulte "Proteção de senha do MQCSP" na página 32.

IDs de Usuário

Quando você cria IDs de usuário para aplicativos clientes, os IDs de usuário não devem ser maiores que o comprimento máximo permitido. Não se deve usar os IDs de usuário reservados UNKNOWN e NOBODY. Se o servidor ao qual o cliente se conectar for um servidor IBM MQ for Windows, você deverá escapar o uso do sinal de arroba, @. O comprimento permitido de IDs de usuários é dependente da plataforma utilizada para o servidor:

- Linux z/OS AIX No z/OS, AIX and Linux, o comprimento máximo de um ID doe usuário é de 12 caracteres.
- IBM i No IBM i, o comprimento máximo de um ID do usuário é 10 caracteres.
- Windows No Windows, se tanto o IBM MQ MQI client quanto o servidor IBM MQ estiverem no Windows e o servidor tiver acesso ao domínio no qual o ID de usuário cliente é definido, o comprimento máximo de um ID de usuário será de 20 caracteres. No entanto, se o servidor IBM MQ não for um servidor Windows, o ID do usuário será truncado para 12 caracteres.
- Se você usar a estrutura MQCSP para passar credenciais, o comprimento máximo de um ID do usuário será 1024 caracteres. O ID do usuário da estrutura MQCSP não pode ser usado para contornar o comprimento máximo de ID de usuário usado por IBM MQ para autorização. Para obter mais informações sobre a estrutura MQCSP, veja "Identificando e autenticando usuários usando a estrutura MQCSP" na página 327.

Em sistemas AIX and Linux o padrão é que IDs de usuário são usados para autenticação, e grupos são usados para autorização. No entanto, é possível configurar esses sistemas para autorização com relação a IDs de usuário. Para obter mais informações, consulte "Permissões baseadas em usuário do OAM no AIX and Linux" na página 359. Os sistemas Windows podem usar IDs de usuário para autenticação e autorização e grupos para autorização.

Se você criar contas de serviço, sem prestar atenção em grupos, e autorizar todos os IDs de usuário diferentes, cada usuário poderá acessar as informações dos outros usuários.

IDs de usuário restritos

Os IDs de usuário UNKNOWN e o grupo NOBODY têm significados especiais para IBM MQ. A criação de um ID de usuário no sistema operacional chamado UNKNOWN ou um grupo chamado NOBODY pode ter resultados indesejados.

IDs de usuário ao se conectar a um servidor IBM MQ for Windows

Windows

Um servidor IBM MQ for Windows não suportará a conexão de um IBM MQ MQI client se o cliente estiver sendo executado sob um ID do usuário que contém o caractere @, por exemplo, `abc@d`. O código de retorno para a chamada MQCONN no cliente é MQRC_NOT_AUTHORIZED.

No entanto, é possível especificar o ID do usuário usando dois caracteres @, por exemplo, `abc@@d`. Usar o formato `id@domain` é a prática preferencial para assegurar que o ID do usuário seja resolvido no domínio correto de forma consistente; portanto, `abc@@d@domain`.

Planejando a autorização

Planeje os usuários que terão autoridade administrativa e planeje como autorizar os usuários de aplicativos a usarem apropriadamente os objetos do IBM MQ, incluindo a conexão a partir de um IBM MQ MQI client.

Deve ser concedido acesso a indivíduos ou aplicações para usar o IBM MQ. O acesso que eles requerem depende das funções que eles realizam e das tarefas que eles precisam executar. A autorização no IBM MQ pode ser subdividida em duas categorias principais:

- Autorização para executar operações administrativas
- Autorização para aplicativos usarem o IBM MQ

As classes de operação são controladas pelo mesmo componente, e a um individual pode ser concedida a autoridade para executar ambas as categorias de operação.

Os tópicos a seguir fornecem informações adicionais sobre áreas específicas de autorização que deve ser consideradas:

Autoridade para administrar o IBM MQ

Os administradores do IBM MQ precisam de autoridade para executar várias funções. Essa autoridade é obtida de diferentes maneiras em diferentes plataformas.

Os administradores do IBM MQ precisam de autoridade para:

- Emitir comandos para administrar o IBM MQ.
- Windows Linux Use o IBM MQ Explorer.
- z/OS Usar os painéis de operações e de controle no z/OS.
- z/OS Usar o programa utilitário do IBM MQ, CSQUTIL, no z/OS.
- z/OS Acessar os conjuntos de dados do gerenciador de filas no z/OS.

Para obter mais informações, consulte o tópico apropriado para seu sistema operacional.

ALW *Autoridade para administrar o IBM MQ em sistemas AIX, Linux, and Windows*

Um administrador do IBM MQ é um membro do grupo mqm. Este grupo tem acesso a todos os recursos do IBM MQ e pode emitir comandos de controle do IBM MQ. Um administrador pode conceder autoridades específicas para outros usuários.

Para ser um administrador do IBM MQ em sistemas AIX, Linux, and Windows, um usuário deve ser um membro do *grupo mqm*. Esse grupo é criado automaticamente quando você instala o IBM MQ. Para permitir que os usuários emitam comandos de controle, deve-se incluí-los no grupo mqm. Isso inclui o usuário raiz no AIX and Linux.

Os usuários que não são membros do grupo mqm podem ser receber privilégios administrativos, mas eles não são capazes de emitir comandos de controle do IBM MQ e estão autorizados a executar apenas os comandos para os quais tiverem recebido acesso.

Além disso, em sistemas Windows , as contas SYSTEM e de Administrador têm acesso total aos recursos do IBM MQ

Todos os membros do grupo mqm têm acesso a todos os recursos do IBM MQ no sistema, incluindo ser capazes de administrar qualquer gerenciador de fila em execução no sistema. Esse acesso pode ser revogado somente com a remoção de um usuário do grupo mqm. Nos sistemas Windows, os membros do grupo de Administradores também têm acesso a todos os recursos do IBM MQ.

Os administradores podem usar o comando de controle **runmqsc** para emitir comandos IBM MQ Script (MQSC). Quando **runmqsc** é utilizado no modo indireto para enviar comandos MQSC para um gerenciador de fila remoto, cada comando será encapsulado dentro de um comando PCF Escape. Os administradores

devem ter as autoridades requeridas para os comandos MQSC serem processados pelo gerenciador de fila remoto.

O IBM MQ Explorer emite comandos PCF (formato de comando programável) para executar tarefas de administração. Os administradores não requerem autoridades adicionais para usar o IBM MQ Explorer para administrar um gerenciador de filas no sistema local. Quando o IBM MQ Explorer é usado para administrar um gerenciador de filas em outro sistema, os administradores devem ter as autoridades necessárias para que os comandos PCF sejam processados pelo gerenciador de filas remoto.

Para obter mais informações sobre as verificações de autoridade efetuadas quando os comando de PCF e MQSC são processados, consulte os seguintes tópicos:

- Para comandos que operam em gerenciadores de filas, filas, canais, processos, listas de nomes e objetos de informações sobre autenticação, consulte "Autorização para aplicativos usarem o IBM MQ" na página 98.
- Para comandos que operam em canais, inicializadores de canais, listeners e clusters, consulte Segurança de canal.
- z/OS Para comandos MQSC que são processados pelo servidor de comando no IBM MQ for z/OS, consulte "Command security and command resource security on z/OS" na página 96.

Para obter mais informações sobre a autoridade que você precisa para administrar os sistemas IBM MQ for AIX, Linux, and Windows, consulte as informações relacionadas.

IBM i *Autoridade para administrar o IBM MQ no IBM i*

Para ser um administrador do IBM MQ no IBM i, deve-se ser um membro do *grupo QMQMADM*. Esse grupo tem propriedades semelhantes às do grupo mqm em sistemas AIX, Linux, and Windows. Em particular, o grupo QMQMADM é criado ao se instalar o IBM MQ for IBM i, e os membros do grupo QMQMADM possuem acesso a todos os recursos do IBM MQ no sistema. Você também tem acesso a todos os recursos do IBM MQ se tiver a autoridade *ALLOBJ.

Os administradores podem usar comandos CL para administrar o IBM MQ.Um desses comandos de controle é GRTMQMAUT, que é utilizado para conceder autoridades a outros usuários. Outro comando, STRMQMMQSC, permite que um administrador emita comandos MQSC para um gerenciador de fila local.

Existem dois grupos do comando CL fornecidos pelo IBM MQ for IBM i:

Grupo 1
Para emitir um comando desta categoria, um usuário deve ser membro do grupo QMQMADM ou ter autoridade de *ALLOBJ. GRTMQMAUT e STRMQMMQSC pertencem a esta categoria, por exemplo.

Grupo 2
Para emitir um comando desta categoria, um usuário não precisa ser membro do grupo QMQMADM ou ter autoridade de *ALLOBJ. Em vez disso, são necessários dois níveis de autoridade:

- O usuário requer a autoridade do IBM i para usar o comando. Esta autoridade é concedida usando o comando GRTOBJAUT.
- O usuário requer a autoridade do IBM MQ para acessar qualquer objeto do IBM MQ associado ao comando. Esta autoridade é concedida usando o comando GRTMQMAUT.

Os exemplos a seguir mostram os comandos neste grupo:

- CRTMQMQ, Criar Fila do MQM
- CHGMQMPRC, Alterar Processo do MQM
- DLTMQMNL, Excluir Liste de Nomes do MQM
- DSPMQMAUTI, Exibir Informações de Autenticação do MQM
- CRTMQMCHL, Criar canal do MQM

Para obter mais informações sobre este grupo de comandos, consulte "Autorização para aplicativos usarem o IBM MQ" na página 98.

Para obter uma lista completa de comandos do grupo 1 e grupo 2, veja “Autoridades de acesso para objetos do IBM MQ no IBM i” na página 165

Para obter mais informações sobre a autoridade que você precisa para administrar o IBM MQ no IBM i, consulte Administrando o IBM i .

z/OS *Authority to administer IBM MQ on z/OS*

This collection of topics describes various aspects of the authority you need to administer IBM MQ for z/OS.

z/OS *Authority checks on z/OS*

IBM MQ for z/OS uses the System Authorization Facility (SAF) to route requests for authority checks to an external security manager (ESM) such as the z/OS Security Server Resource Access Control Facility (RACF). IBM MQ does no authority checks of its own.

It is assumed that you are using RACF as your ESM. If you are using a different ESM, you might need to interpret the information provided for RACF in a way that is relevant to your ESM.

You can specify whether you want authority checks turned on or off for each queue manager individually or for every queue manager in a queue sharing group. This level of control is called *subsystem security*. If you turn subsystem security off for a particular queue manager, no authority checks are carried out for that queue manager.

If you turn subsystem security on for a particular queue manager, authority checks can be performed at two levels:

Queue sharing group level security
Authority checks use RACF profiles that are shared by all queue managers in the queue sharing group. This means that there are fewer profiles to define and maintain, making security administration easier.

Queue manager level security
Authority checks use RACF profiles specific to the queue manager.

You can use a combination of queue sharing group and queue manager level security. For example, you can arrange for profiles specific to a queue manager to override those of the queue sharing group to which it belongs.

Subsystem security, queue sharing group level security, and queue manager level security are turned on or off by defining *switch profiles*. A switch profile is a normal RACF profile that has a special meaning to IBM MQ.

z/OS *Command security and command resource security on z/OS*

Command security relates to the authority to issue a command; command resource authority relates to the authority to perform an operation on a resource. Both are implemented y using RACF classes.

Authority checks are carried out when an IBM MQ administrator issues an MQSC command. This is called *command security*.

To implement command security, you must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. The name of a profile for command security contains the name of an MQSC command.

Some MQSC commands perform an operation on an IBM MQ resource, such as the DEFINE QLOCAL command to create a local queue. When an administrator issues an MQSC command, authority checks are carried out to determine whether the requested operation can be performed on the resource specified in the command. This is called *command resource security*.

To implement command resource security, you must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. The name of a profile for command resource security contains the name of an IBM MQ resource and its type (QUEUE, PROCESS, NAMELIST, TOPIC, AUTHINFO, or CHANNEL).

Command security and command resource security are independent. For example, when an administrator issues the command:

```
DEFINE QLOCAL(MOON.EUROPA)
```

the following authority checks are performed:

- Command security checks that the administrator is authorized to issue the DEFINE QLOCAL command.
- Command resource security checks that the administrator is authorized to perform an operation on the local queue called MOON.EUROPA.

Command security and command resource security can be turned on or off by defining switch profiles.

z/OS *MQSC commands and the system command input queue on z/OS*

Use this topic to understand how the command server processes MQSC commands directed to the system command input queue on z/OS.

Command security and command resource security are also used when the command server retrieves a message containing an MQSC command from the system command input queue. The user ID that is used for the authority checks is the one found in the *UserIdentifier* field in the message descriptor of the message containing the MQSC command. This user ID must have the required authorities on the queue manager where the command is processed. For more information about the *UserIdentifier* field and how it is set, see Message context.

Messages containing MQSC commands are sent to the system command input queue in the following circumstances:

- The operations and control panels send MQSC commands to the system command input queue of the target queue manager. The MQSC commands correspond to the actions you choose on the panels. The *UserIdentifier* field in each message is set to the TSO user ID of the administrator.
- The COMMAND function of the IBM MQ utility program, CSQUTIL, sends the MQSC commands in the input data set to the system command input queue of the target queue manager. The COPY and EMPTY functions send DISPLAY QUEUE and DISPLAY STGCLASS commands. The *UserIdentifier* field in each message is set to the job user ID.
- The MQSC commands in the CSQINPX data sets are sent to the system command input queue of the queue manager to which the channel initiator is connected. The *UserIdentifier* field in each message is set to the channel initiator address space user ID.

  No authority checks are performed when MQSC commands are issued from the CSQINP1 and CSQINP2 data sets. You can control who is allowed to update these data sets using RACF data set protection.
- Within a queue sharing group, a channel initiator might send START CHANNEL commands to the system command input queue of the queue manager to which it is connected. A command is sent when an outbound channel that uses a shared transmission queue is started by triggering. The *UserIdentifier* field in each message is set to the channel initiator address space user ID.
- An application can send MQSC commands to a system command input queue. By default, the *UserIdentifier* field in each message is set to the user ID associated with the application.
- On AIX, Linux, and Windows systems, the **runmqsc** control command can be used in indirect mode to send MQSC commands to the system command input queue of a queue manager on z/OS. The *UserIdentifier* field in each message is set to the user ID of the administrator who issued the **runmqsc** command.

z/OS *Access to the queue manager data sets on z/OS*

IBM MQ for z/OS administrators need authority to access the queue manager data sets. Use this topic to understand which data sets need RACF protection.

These data sets include:

- The data sets referred to by CSQINP1, CSQINP2, and CSQINPT in the started task procedure of the queue manager.

- The queue manager's page sets, active log data sets, archive log data sets, and bootstrap data sets (BSDSs)
- The data sets referred to by CSQXLIB and CSQINPX in the channel initiator's started task procedure

You must protect the data sets so that no unauthorized user can start a queue manager or gain access to any queue manager data. To do this, use RACF data set protection.

Autorização para aplicativos usarem o IBM MQ

Quando os aplicativos acessam objetos, os IDs de usuário associados aos aplicativos precisam de autoridade apropriada.

Os aplicativos podem acessar os seguintes objetos do IBM MQ, emitindo chamadas de MQI:

- Gerenciadores de filas
- Filas
- Processos
- Listas de Nomes
- tópicos

Os aplicativos também podem usar comando de PCF para administrar objetos do IBM MQ. Quando o comando PCF é processado, ele usa o contexto de autoridade do ID do usuário que envia a mensagem de PCF.

Os aplicativos, nesse contexto, incluem aqueles gravados por usuários e fornecedorese aqueles fornecidos com o IBM MQ for z/OS

z/OS Os aplicativos fornecidos com o IBM MQ for z/OS incluem:

- Os painéis de operações e controle
- O programa utilitário IBM MQ, CSQUTIL
- O utilitário CSQUDLQH (Dead Letter Queue Handler)

Os aplicativos que usam IBM MQ classes for Java, IBM MQ classes for JMS, IBM MQ classes for .NET ou o Message Service Clients for C/C++ e .NET usam o MQI indiretamente.

Os MCAs também emitem chamadas de MQI e os IDs do usuário associados a MCAs necessitam de autoridade para acessar esses objetos do IBM MQ. Para obter mais informações sobre esses IDs do usuário e as autoridades que eles requerem, consulte a seção “Autorização de canal” na página 120.

z/OS No z/OS, os aplicativos também podem usar comandos MQSC para acessar esses objetos do IBM MQ, porém, a segurança do comando e a segurança do recurso do comando fornecem as verificações de autoridade nesses casos. z/OS Para obter mais informações, consulte “Command security and command resource security on z/OS” na página 96 e “MQSC commands and the system command input queue on z/OS” na página 97.

IBM i No IBM i, um usuário que emite um comando CL no Grupo 2 pode requerer de autoridade para acessar um objeto do IBM MQ associado ao comando. Para obter informações adicionais, consulte “Quando as Verificações de Autoridade são Executadas” na página 98.

Quando as Verificações de Autoridade são Executadas

As verificações de autoridade são executadas quando um aplicativo tenta acessar um gerenciador de filas, uma fila, um processo ou uma lista de nomes.

No IBM i, as verificações de autoridade também podem ser realizadas quando um usuário emite um comando CL no Grupo 2 que acessa qualquer um destes objetos do IBM MQ. As verificações são executadas nas seguintes situações:

Quando um aplicativo estabelece conexão com um gerenciador de fila usando uma chamada MQCONN ou MQCONNX

O gerenciador de fila solicita o sistema operacional do ID do usuário associado ao aplicativo. Então, verifica se o ID do usuário tem autorização para estabelecer conexão com ele e o retém para verificações futuras.

Os usuários não têm que efetuar sign on para o IBM MQ. O IBM MQ assume que os usuários estão conectados ao sistema operacional subjacente e que foram autenticados por ele.

Quando um aplicativo abre um objeto do IBM MQ usando uma chamada MQOPEN ou MQPUT1

Todas as verificações de autoridade são executadas quando um objeto é aberto, não ao ser acessado posteriormente. Por exemplo, as verificações de autoridade são executadas quando um aplicativo abre uma fila. Elas não são executadas quando o aplicativo coloca mensagens na fila ou recebe mensagens da fila.

Quando um aplicativo abre um objeto, ele especifica os tipos de operação de que necessita executar nele. Por exemplo, um aplicativo pode abrir uma fila para consultar e obter mensagens, mas não para colocar mensagens nela. Para cada tipo de operação, o gerenciador de fila verifica se o ID do usuário associado ao aplicativo tem a autoridade para executar essa operação.

Quando um aplicativo abre uma fila, as verificações de autoridade são executadas no objeto nomeado no campo `ObjectName` do descritor de objeto. O campo `ObjectName` é usado nas chamadas `MQOPEN` ou `MQPUT1`. Se o objeto for uma fila de alias ou uma definição de fila remota, as verificações de autoridade serão executadas no próprio objeto. Elas não são executadas na fila para a qual a fila de alias ou a definição de fila remota é resolvida. Isso significa que o usuário não precisa de permissão para acessá-la. Limite a autoridade para criar filas a usuários privilegiados. Se você não fizer isto, os usuários podem efetuar bypass do controle de acesso normal simplesmente criando um alias.

Um aplicativo pode referenciar uma fila remota explicitamente. Ele configura os campos `ObjectName` e `ObjectQMgrName` no descritor de objetos para os nomes da fila remota e o gerenciador de filas remotas. As verificações de autoridade são executadas na fila de transmissão com o mesmo nome que o gerenciador de filas remotas:

- z/OS No z/OS, é feita uma verificação no perfil da fila RACF que corresponde ao nome do gerenciador de filas remotas e é executada se essa fila de transmissão for definida localmente.
- Multi No Multiplataformas, uma verificação será feita com relação ao perfil RQMNAME que corresponde ao nome do gerenciador de filas remotas, se o armazenamento em cluster estiver sendo utilizado

Um aplicativo pode fazer referência a uma fila de clusters explicitamente, definindo o campo `ObjectName` no descritor de objeto com o nome da fila de clusters. As verificações de autoridade são executadas na fila de transmissão do cluster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

A autoridade para uma fila dinâmica é baseada na fila modelo da qual se deriva, mas não é necessariamente a mesma; consulte nota 1.

O ID do usuário que o Gerenciador de Filas utiliza para as verificações de autoridade é obtido do sistema operacional. O ID do usuário é obtido quando o aplicativo estabelece conexão com o gerenciador de filas. Um aplicativo adequadamente autorizado pode emitir uma chamada `MQOPEN` especificando um ID do usuário alternativo; as verificações de controle de acesso são então feitas no ID do usuário alternativo. Usar um ID de usuário alternativo não muda o ID de usuário associado ao aplicativo, apenas o ID que é usado para verificações de controle de acesso.

Quando um aplicativo é subscrito para um tópico usando uma chamada MQSUB

Quando um aplicativo é subscrito para um tópico, ele especifica o tipo de operação que precisa executar. Ele estará criando uma assinatura, alterando uma assinatura existente ou continuando uma assinatura existente sem mudá-la. Para cada tipo de operação, o gerenciador de filas verifica se o ID do usuário que está associado ao aplicativo possui a autoridade para executar a operação.

Quando um aplicativo é subscrito para um tópico, as verificações de autoridade são executadas com relação a objetos de tópicos que estão localizados na árvore de tópicos. Os objetos do tópico estão em, ou acima do ponto na árvore de tópicos na qual o aplicativo foi inscrito. As verificações

de autoridade podem envolver verificações em mais de um objeto de tópico. O ID do usuário que o Gerenciador de Filas utiliza para as verificações de autoridade é obtido do sistema operacional. O ID do usuário é obtido quando o aplicativo estabelece conexão com o gerenciador de filas.

O gerenciador de filas executa verificações de autoridade em filas de assinantes, mas não em filas gerenciadas.

Quando um aplicativo exclui uma fila dinâmica permanente usando uma chamada MQCLOSE

A manipulação de objetos especificada na chamada MQCLOSE não é necessariamente a mesma retornada pela chamada MQOPEN que criou a fila dinâmica permanente. Se for diferente, o gerenciador de filas verifica o ID do usuário associado ao aplicativo que emitiu a chamada MQCLOSE. Ele verifica se o ID de usuário tem autorização para excluir a fila.

Quando um aplicativo que fecha uma assinatura para removê-la não a criou, a autoridade apropriada é requerida para removê-la.

Quando um comando PCF que opera em um objeto do IBM MQ é processado pelo servidor de comandos

Essa regra inclui o caso em que um comando PCF opera em um objeto de informações sobre autenticação.

O ID do usuário utilizado para as verificações de autoridade é o encontrado no campo UserIdentifier , no descritor de mensagens do comando PCF. Esse ID do usuário deve ter as autoridades requeridas no gerenciador de fila em que o comando é processado. O comando MQSC equivalente encapsulado dentro de um comando PCF Escape é tratado da mesma forma. Para obter mais informações sobre o campo UserIdentifier e como ele está definido, consulte “Contexto da mensagem” na página 101.

IBM i **No IBM i, quando um usuário emite um comando CL no Grupo 2 que opera em um objeto do IBM MQ**

Essa regra inclui o caso em que um comando CL no Grupo 2 opera em um objeto de informações sobre autenticação.

As verificações são feitas para determinar se o usuário tem a autoridade para operar em um objeto do IBM MQ associado ao comando. As verificações são executadas, a menos que o usuário seja um membro do grupo QMQMADM ou tenha autoridade *ALLOBJ. A autoridade necessária depende do tipo de operação que o comando realiza sobre o objeto. Por exemplo, o comando **CHGMQMQ**, Change MQM Queue, requer autoridade para mudar atributos da fila especificada pelo comando. Em contraste, o comando **DSPMQMQ**, Display MQM Queue, requer a autoridade de exibir os atributos da fila especificada pelo comando.

Muitos comandos operam sobre mais de um objeto. Por exemplo, para emitir o comando **DLTMQMQ**, Delete MQM Queue, as seguintes autoridades são necessárias:

- A autoridade de conectar ao gerenciador de filas especificado pelo comando
- A autoridade de excluir a fila especificada pelo comando

Alguns comandos não operam sobre objeto algum. Nesse caso, o usuário requer apenas a autoridade IBM i para emitir um desses comandos **STRMQMLSR**, Start MQM Listener, é um exemplo de tal comando.

Alternar autoridade do usuário

Quando um aplicativo abre um objeto ou assina um tópico, o aplicativo pode fornecer um ID de usuário na chamada MQOPEN, MQPUT1 ou MQSUB. Ele pode solicitar ao gerenciador de filas para usar esse ID de usuário para verificações de autoridade, em vez daquele associado ao aplicativo.

O aplicativo será bem-sucedido ao abrir o objeto somente se as duas condições a seguir forem atendidas:

- O ID do usuário associado ao aplicativo tem autoridade para fornecer um ID de usuário diferente para verificações de autoridade. Diz-se que o aplicativo tem *autoridade de usuário alternativo*.
- O ID do usuário fornecido pelo aplicativo possui a autoridade para abrir o objeto para os tipos de operações solicitadas ou para assinar o tópico.

Contexto da mensagem

As informações sobre *contexto da mensagem* permitem que o aplicativo que recupera uma mensagem descubra o emissor dela. As informações ficam retidas em campos no descritor de mensagens e os campos são divididos em três partes lógicas

Essas partes são as seguintes:

contexto de identidade
Estes campos contêm informações sobre o usuário do aplicativo que colocou a mensagem na fila.

contexto de origem
Estes campos contêm informações sobre o aplicativo em si e quando foi que a mensagem foi colocada na fila.

contexto do usuário
Estes campos contêm propriedades de mensagens que os aplicativos utilizam para selecionar as mensagens que o gerenciador de filas deve entregar.

Quando um aplicativo coloca uma mensagem em uma fila, pode solicitar que o gerenciador de fila gere as informações de contexto na mensagem. Esta é a ação padrão. Alternativamente, ele pode especificar que os campos de contexto não contenham informações. O ID do usuário associado a um aplicativo não requer autoridade especial para nenhum desses.

Um aplicativo pode definir os campos de contexto de identidade em uma mensagem, permitindo que o gerenciador de fila gere o contexto de origem ou pode definir todos os campos de contexto. Pode também passar os campos de contexto de identidade de uma mensagem recuperada para uma mensagem que esteja colocando na fila ou passar todos os campos de contexto. Porém, o ID do usuário associado a um aplicativo requer autoridade para definir ou passar as informações de contexto. Um aplicativo especifica se pretende definir ou passar informações de contexto quando abrir a fila na qual está para colocar mensagens e sua autoridade é verificada nesse momento.

A seguir, uma descrição breve de cada campo de contexto:

Contexto de Identidade

UserIdentifier
O ID do usuário associado ao aplicativo que colocou a mensagem. Se o gerenciador de fila definir este campo, ele será definido com o ID obtido do sistema operacional de quando o aplicativo estabelece conexão com o gerenciador de fila.

AccountingToken
As informações podem ser utilizadas para cobrar o trabalho feito como resultado da mensagem.

ApplIdentityData
Se o ID do usuário associado ao aplicativo tiver autoridade para definir os campos de contexto de identidade ou todos os campos de contexto, o aplicativo poderá definir esse campo com qualquer valor relacionado à identidade. Se o gerenciador de fila definir esse campo, será deixado em branco.

Contexto de origem

PutApplType
O tipo do aplicativo que coloca a mensagem; uma transação do CICS, por exemplo.

PutApplName
O nome do aplicativo que coloca a mensagem.

PutDate
A data em que a mensagem foi colocada.

PutTime
A hora em que a mensagem foi colocada.

ApplOriginData
Se o ID do usuário associado ao aplicativo tiver autoridade para definir todos os campos de contexto, o aplicativo poderá definir esse campo com qualquer valor relacionado à origem. Se o gerenciador de fila definir esse campo, será deixado em branco.

Contexto de Usuário

Os seguintes valores são suportados para **MQINQMP** ou **MQSETMP**:

MQPD_USER _CONTEXT

A propriedade é associada com o contexto do usuário.

Não é necessária nenhuma autorização especial para poder definir uma propriedade associada ao contexto do usuário utilizando a chamada MQSETMP.

Em um gerenciador de filas V7.0 ou subsequente, uma propriedade associada ao contexto de usuário é salva conforme descrito para MQOO_SAVE_ALL_CONTEXT. Um MQPUT com MQOO_PASS_ALL_CONTEXT especificado faz com que a propriedade seja copiada do contexto salvo para a nova mensagem.

MQPD_NO_CONTEXT

A propriedade não é associada com um contexto de mensagem.

Um valor não reconhecido é rejeitado com um MQRC_PD_ERROR. O valor inicial deste campo é **MQPD_NO_CONTEXT**.

Para obter uma descrição detalhada de cada um dos campos de contexto, consulte MQMD - Descritor de mensagens. Para obter mais informações sobre como usar o contexto da mensagem, consulte Contexto da Mensagem.

IBM i ALW *Autoridade para trabalhar com objetos do IBM MQ em sistemas IBM i, AIX, Linux, and Windows*

O componente de serviço de autorização fornecido com o IBM MQ é chamado de *gerenciador de autoridade de objeto* (OAM). Ele fornece controle de acesso por meio de verificações de autenticação e autorização.

Autenticação.

A verificação de autenticação executada pelo OAM fornecido com o IBM MQ é básico, e é executada apenas em circunstâncias específicas. Seu objetivo não é atender aos requisitos rígidos esperados em um ambiente altamente seguro.

O OAM realiza sua verificação de autenticação quando um aplicativo se conecta a um Gerenciador de Filas e as condições a seguir são verdadeiras:

- Se uma estrutura MQCSP tiver sido fornecida pelo aplicativo de conexão, e
- Ao atributo *AuthenticationType* na estrutura MQCSP é dado o valor MQCSP_AUTH_USER_ID_AND_AND_PWD, e
- O valor CHCKLOCL ou CHKCCLNT no objeto AUTHINFO configurado não é 'NONE'

As etapas de autenticação no OAM validam a senha usando os serviços do sistema operacional, que podem ter sido configurados para realizar verificações adicionais, como garantir que o nome de usuário não tenha tido muitas tentativas incorretas de teste de senha.

É possível que mecanismos de autenticação alternativos sejam usados se você escrever um novo componente de serviço de autorização ou obtiver um de um fornecedor.

Autorização.

As verificações de autorização são abrangentes e seu objetivo é atender a requisitos mais normais.

As verificações de autorização são executadas quando um aplicativo emite uma chamada MQI para acessar um gerenciador de filas, uma fila, um processo, um tópico ou uma lista de nomes. Elas também são executadas em outros momentos, por exemplo, quando um comando está sendo executado pelo Servidor de Comandos.

Em sistemas IBM i IBM i, AIX, Linux, and Windows, o *serviço de autorização* fornecerá o controle de acesso quando um aplicativo emitir uma chamada MQI para acessar um objeto do IBM MQ que seja

um gerenciador de filas, uma fila, um processo, um tópico ou uma lista de nomes. Isso inclui verificações da autoridade do usuário alternativo e a autoridade para definir ou passar informações de contexto.

Windows No Windows, o OAM fornece aos membros do grupo de Administradores a autoridade para acessar todos os objetos do IBM MQ, mesmo quando o UAC está ativado. Além disso, em sistemas Windows , a conta SYSTEM tem acesso total aos recursos do IBM MQ

O serviço de autorização também fornece verificações de autoridade quando um comando de PCF opera em um desses objetos do IBM MQ ou em um objeto de informações sobre autenticação. O comando MQSC equivalente encapsulado dentro de um comando PCF Escape é tratado da mesma forma.

IBM i No IBM i, a menos que o usuário seja um membro do grupo QMQMADM ou tenha autoridade *ALLOBJ, o serviço de autorização também fornece verificações de autoridade quando um usuário emite um comando CL no Grupo 2 que opera sobre qualquer um destes objetos do IBM MQ ou um objeto de informações sobre autenticação.

O serviço de autorização é um *serviço instalável* , que significa que é implementado por um ou mais *componentes de serviços instaláveis*. Cada componente é chamado por uma interface documentada. Isto permite que usuários e fornecedores forneçam componentes para aumentar ou substituir os fornecidos pelos produtos IBM MQ.

O componente de serviço de autorização fornecido com o IBM MQ é chamado de gerenciador de autoridade de objeto (OAM). O OAM é ativado automaticamente para cada gerenciador de fila criado.

O OAM mantém uma lista de controle de acesso (ACL) para cada objeto do IBM MQ para o qual ele está controlando o acesso. Em sistemas AIX and Linux, somente IDs de grupos podem aparecer em uma ACL. Isto significa que todos os membros de um grupo possuem as mesmas autoridades. Em sistemas IBM i IBM i e Windows, os IDs do usuário e os IDs do grupo podem aparecer em uma ACL. Isto significa as autoridades podem ser concedidas para usuários individuais e grupos.

Uma limitação de 12 caracteres aplica-se ao grupo e ao ID do usuário. As plataformas UNIX geralmente restringem o comprimento de um ID do usuário a 12 caracteres. O AIX e o Linux emitiam esse limite, mas IBM MQ continua a observar uma restrição de 12 caracteres em todas as plataformas UNIX. Se você usar um ID do usuário com mais de 12 caracteres, o IBM MQ substitui-o com o valor de "UNKNOWN". Não defina um ID do usuário com um valor de "UNKNOWN".

O OAM pode autenticar um usuário e alterar os campos de contexto de identidade adequados. Você ativa isto especificando uma estrutura MQCSP (connection security parameters) em uma chamada MQCONNX. A estrutura é passada para a função de Autenticar Usuário OAM (MQZ_AUTHENTICATE_USER), que define campos de contexto de identidade adequados. Se uma conexão do MQCONNX a partir de um cliente IBM MQ, as informações no MQCSP são fluídas para o gerenciador de filas ao qual o cliente está se conectando através do canal de conexão do cliente e de conexão do servidor. Se as saídas de segurança estiverem definidas neste canal, a MQCSP é passada para cada saída de segurança e pode ser alterada pela saída. As saídas de segurança também podem criar a MQCSP. Para obter mais detalhes do uso das saídas de segurança neste contexto, consulte Programas de saída de segurança do canal.

Aviso: Em alguns casos, a senha em uma estrutura MQCSP para um aplicativo cliente será enviada através de uma rede em texto simples. Para assegurar que as senhas do aplicativo cliente sejam protegidas adequadamente, consulte IBM MQProteção de senha CSP.

Em sistemas AIX, Linux, and Windows, o comando de controle **setmqaut** concede e revoga autoridades e é usado para manter as ACLs. Por exemplo, o comando:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER +browse +get
```

permite que membros do grupo VOYAGER procurem mensagens na fila MOON.EUROPA que é de propriedade do gerenciador de fila JUPITER. Permite também que os membros obtenham mensagens da fila. Para revogar estas autoridades posteriormente, insira o seguinte comando:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER -browse -get
```

O comando:

```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +put
```

permite que membros do grupo VOYAGER coloquem mensagens em qualquer fila com um nome que comece com os caracteres MOON.. MOON.* é o nome de um perfil genérico..Um *perfil genérico* permite conceder autoridades para um conjunto de objetos usando um único comando **setmqaut** .

O comando de controle **dspmqaut** está disponível para exibir as autoridades atuais que um usuário ou um grupo tem para um objeto especificado.O comando de controle **dmpmqaut** também está disponível para exibir as autoridades atuais associadas aos perfis genéricos.

IBM i No IBM i, um administrador usa o comando CL GRTMQMAUT para conceder autoridades e o comando CL RVKMQMAUT para revogar autoridades. Perfis genéricos também podem ser utilizados.Por exemplo, o comando CL:

```
GRTMQMAUT MQMNAME(JUPITER) OBJTYPE(*Q) OBJ('MOON.*') USER(VOYAGER) AUT(*PUT)
```

fornece a mesma função que o exemplo anterior de um comando **setmqaut**. Ele permite que os membros do grupo VOYAGER coloquem mensagens em qualquer fila com um nome que inicia com os caracteres MOON.

IBM i O comando CL DSPMQMAUT exibe as autoridades atuais que o usuário ou grupo tem para um determinado objeto. Os comandos CL WRKMQMAUT e WRKMQMAUTD também estão disponíveis para trabalhar com as autoridades atualmente associadas a objetos e perfis genéricos.

Se você não desejar nenhuma verificação de autoridade, por exemplo, em um ambiente de teste, poderá desativar o OAM.

Multi *Usando PCF para acessar comandos OAM*

Nos sistemas IBM i, AIX, Linux, and Windows, é possível usar comandos PCF para acessar comandos de administração OAM.

Os comando de PCF e seus comandos OAM equivalentes são os seguintes:

Tabela 8. Comandos PCF e seus comandos OAM equivalentes

comando PCF	**Comando OAM**
Consultar Registros de Autoridade	dmpmqaut
Solicitar Autoridade de Entidade	dspmqaut
Configurar Registro de Autoridade	setmqaut
Excluir Registro de Autoridade	setmqaut com opção -remove

Os comandos **setmqaut** e **dmpmqaut** são restritos a membros do grupo mqm. Os comandos de PCF equivalentes podem ser executados por usuários em qualquer grupo que tenha recebido as autoridades dsp e chg no gerenciador de filas.

Para obter mais informações sobre como usar esses comandos, consulte Introdução aos formatos de comando programável.

z/OS *Authority to work with IBM MQ objects on z/OS*

On z/OS, there are seven categories of authority check associated with calls to the MQI. You must define certain RACF profiles and give appropriate access to these profiles. Use the *RESLEVEL* profile to control how many users IDs are checked.

The seven categories of authority check associated with calls to the MQI:

Connection security
The authority checks that are performed when an application connects to a queue manager

Queue security
The authority checks that are performed when an application opens a queue or deletes a permanent dynamic queue

Process security
The authority checks that are performed when an application opens a process object

Namelist security
The authority checks that are performed when an application opens a namelist object

Alternate user security
The authority checks that are performed when an application requests alternate user authority when opening an object

Context security
The authority checks that are performed when an application opens a queue and specifies that it intends to set or pass the context information in the messages it puts on the queue

Topic security
The authority checks that are performed when an application opens a topic

Each category of authority check is implemented in the same way that command security and command resource security are implemented. You must define certain RACF profiles and give the necessary groups and user IDs access to these profiles at the required levels. For queue security, the level of access determines the types of operation the application can perform on a queue. For context security, the level of access determines whether the application can:

- Pass all the context fields
- Pass all the context fields and set the identity context fields
- Pass and set all the context fields

Each category of authority check can be turned on or off by defining switch profiles.

All the categories, except connection security, are known collectively as *API-resource security*.

By default, when an API-resource security check is performed as a result of an MQI call from an application using a batch connection, only one user ID is checked. When a check is performed as a result of an MQI call from a CICS or IMS application, or from the channel initiator, two user IDs are checked.

By defining a *RESLEVEL profile*, however, you can control whether zero, one, or two users IDs are checked. The number of user IDs that are checked is determined by the user ID associated with the type of connection when an application connects to the queue manager and the access level that user ID has to the RESLEVEL profile. The user ID associated with each type of connection is:

- The user ID of the connecting task for batch connections
- The CICS address space user ID for CICS connections
- The IMS region address space user ID for IMS connections
- The channel initiator address space user ID for channel initiator connections

For more information about the authority to work with IBM MQ objects on z/OS, see “Authority to administer IBM MQ on z/OS” on page 96.

Segurança para o Sistema de Mensagens Remoto

Esta seção trata dos aspectos do sistema de mensagens remoto de segurança.

Você deve fornecer aos usuários a autoridade para usar os recursos do IBM MQ. Isto está organizado de acordo com as ações a serem obtidas em relação aos objetos e definições. Por exemplo:

- Os gerenciadores de fila podem ser iniciados ou parados por usuários autorizados
- Os aplicativos devem se conectar ao gerenciador de filas e ter autoridade para usar as filas

- Os canais de mensagens devem ser criados e controlados pelos usuários autorizados
- Os objetos são mantidos nas bibliotecas e o acesso a essas bibliotecas pode estar restrito

O agente do canal de mensagem em um site remoto deve verificar se a mensagem está sendo entregue originada de um usuário com autoridade para tal neste site remoto. Além disso, como os MCAs podem ser iniciados remotamente, pode ser necessário verificar se os processos remotos que tentam iniciar os seus MCAs estão autorizados a fazê-lo. Existem quatro maneira possíveis para tratar disso:

1. Fazer uso apropriado do atributo PutAuthority da sua definição de canal RCVR, RQSTR ou CLUSRCVR para controlar qual usuário é usado para as verificações de autorização no momento em que as mensagens recebidas são colocadas nas suas filas. Consulte a descrição do comando DEFINE CHANNEL na Referência de Comandos MQSC.
2. Implemente os registros de autenticação de canal para rejeitar as tentativas de conexão indesejadas ou para configurar um valor MCAUSER com base no seguinte: endereço IP remoto, ID do usuário remoto, Nome Distinto (DN) do assunto TLS fornecido ou nome do gerenciador de filas remotas.
3. Implemente a verificação de segurança de *saída de usuário* verificando se o canal de mensagem correspondente está autorizado. A segurança da instalação que hospeda o canal correspondente assegura que todos os usuários estejam corretamente autorizados, de modo que você não precise verificar as mensagens individuais.
4. Implemente o processamento de mensagens de *saída de usuário* para assegurar que as mensagens individuais sejam examinadas para autorização.

IBM i *Segurança de objetos do IBM MQ for IBM i*

Esta seção trata dos aspectos do sistema de mensagens remoto de segurança.

Deve-se fornecer aos usuários a autoridade para usar os recursos do IBM MQ for IBM i. Esta autoridade é organizada de acordo com as ações a serem obtidas em relação aos objetos e definições. Por exemplo:

- Os gerenciadores de fila podem ser iniciados ou parados por usuários autorizados
- Os aplicativos precisam se conectar ao gerenciador de filas, e ter autoridade para fazer uso de filas
- Os canais de mensagens precisam ser criados e controlados pelos usuários autorizados

O agente do canal de mensagens em um site remoto deve verificar se a mensagem que está sendo entregue é derivada de um usuário com autoridade para emitir mensagem neste site remoto. Além disso, como os MCAs podem ser iniciados remotamente, pode ser necessário verificar se os processos remotos que tentam iniciar os seus MCAs estão autorizados a fazê-lo. Existem quatro maneira possíveis para tratar disso:

- Decrete na definição de canal que as mensagens devem conter autoridade *context* aceitável, caso contrário, elas serão descartadas.
- Implemente os registros de autenticação de canal para rejeitar as tentativas de conexão indesejadas ou para configurar um valor MCAUSER com base no seguinte: endereço IP remoto, ID do usuário remoto, Nome Distinto (DN) do TLS ou nome do gerenciador de filas remotas.
- Implemente a verificação de segurança de saída de usuário para assegurar que o canal de mensagens correspondente está autorizado. A segurança da instalação que hospeda o canal correspondente assegura que todos os usuários estejam corretamente autorizados, de modo que você não precise verificar as mensagens individuais.
- Implemente o processamento de mensagens de saída de usuário para assegurar que as mensagens individuais sejam examinadas para autorização.

Aqui estão alguns fatos sobre a maneira como o IBM MQ for IBM i opera em relação à segurança:

- Os usuários são identificados e autenticados pelo IBM i.
- Os serviços do gerenciador de filas, chamados pelos aplicativos, são executados com a autoridade do perfil do usuário do gerenciador de filas, mas no processo do usuário.
- Os serviços do gerenciador de serviços, chamados por comandos do usuário, são executados com a autoridade do perfil do usuário do gerenciador de filas.

Linux AIX *Segurança de objetos no AIX and Linux*

Os usuários de Administração devem ser parte do grupo mqm em seu sistema (incluindo raiz), se este ID usar os comandos de administração do IBM MQ.

É necessário sempre executar amqcrsta como o ID do usuário "mqm".

IDs do Usuário no AIX and Linux

O gerenciador de filas converte todos os identificadores com letras maiúsculas ou com letras maiúsculas e minúsculas em identificadores com letras minúsculas. O gerenciador de filas, então, insere os identificadores de usuário no contexto da parte de uma mensagem, ou verifica sua autorização. As autorizações são, portanto, baseadas apenas em identificadores com letras minúsculas.

Windows *A segurança de objetos em sistemas Windows*

Os usuários de Administração devem ser parte do grupo mqm e do grupo de administradores nos sistemas Windows se este ID usar os comandos de administração do IBM MQ.

Os IDs do usuário nos sistemas Windows

Em sistemas Windows, *se não houver saída de mensagem instalada*, o gerenciador de filas converte qualquer identificador de usuário com letras maiúsculas ou com letras maiúsculas e minúsculas em identificadores com letras minúsculas. O gerenciador de filas, então, insere os identificadores de usuário no contexto da parte de uma mensagem, ou verifica sua autorização. As autorizações são, portanto, baseadas apenas em identificadores com letras minúsculas.

IDs do usuário em sistemas

Plataformas diferentes de sistemas AIX, Linux, and Windows usam caracteres maiúsculos para IDs do usuário em mensagens. Para permitir que os sistemas AIX, Linux, and Windows usem IDs do usuário minúsculos em mensagens, o agente do canal de mensagens (MCA) deve realizar as conversões apropriadas de caracteres alfabéticos.

Para permitir que os sistemas AIX, Linux, and Windows usem IDs do usuário minúsculos em mensagens, as conversões a seguir são realizadas pelo agente do canal de mensagens (MCA) nestas plataformas:

Na extremidade de envio
Os caracteres alfabéticos em todos os IDs do usuário são convertidos em caracteres maiúsculos, se não houver nenhuma saída de mensagem instalada.

Na extremidade de recebimento
Os caracteres alfabéticos em todos os IDs de usuário são convertidos em caracteres minúsculos, se nenhuma saída de mensagem instalada.

As conversões automáticas não serão realizadas se você fornecer uma saída de mensagem no AIX, Linux, and Windows por qualquer outra razão.

Usando um serviço de autorização customizado

O IBM MQ fornece um serviço de autorização instalável. É possível escolher instalar um serviço alternativo.

O componente de serviço de autorização fornecido com o IBM MQ é chamado Gerenciador de Autoridade de Objeto (OAM). Se o OAM não fornecer as instalações de autorização necessárias, será possível gravar seu próprio componente de serviço de autorização. As funções do serviço instalável, que deve ser implementado por um componente de serviço de autorização, são descritas em Informações de referência da interface de serviços instaláveis.

Controle de acesso para clientes

O controle de acesso é baseado nos IDs de usuário. Pode haver muitos IDs de usuário para administrar, e os IDs de usuário podem estar em diferentes formatos. É possível configurar a propriedade MCAUSER do canal de conexão do servidor com um valor de ID do usuário especial para uso pelos clientes.

O controle de acesso em IBM MQ é baseado nos IDs do usuário. O ID do usuário do processo que faz chamadas MQI é normalmente usado. Para os clientes do MQ MQI, o MCA de conexão do servidor faz chamadas MQI em nome de clientes do MQ MQI. É possível selecionar um ID do usuário alternativo para a conexão do servidor MCA para usar para fazer chamadas MQI. O ID do usuário alternativo pode ser associado à estação de trabalho do cliente ou a qualquer coisa escolhida para organizar e controlar o acesso dos clientes. O ID do usuário precisa ter as autoridades necessárias alocadas para ele no servidor para emitir chamadas MQI. Escolher um ID do usuário alternativo é preferível a permitir que clientes façam chamadas MQI com a autoridade da conexão do servidor MCA.

Tabela 9. O ID do usuário usado por um canal de conexão do servidor

ID do usuário	Quando usado
O ID do usuário que é configurado por uma saída de segurança	Usado a menos que seja bloqueado por uma regra **CHLAUTH TYPE(BLOCKUSER)**. Consulte a seção a seguir, "Configurando o ID do usuário em uma saída de segurança" na página 109, para obter mais informações.
O ID do usuário que é configurado por uma regra CHLAUTH	Usado a menos que substituído por uma saída de segurança. Consulte Registros de autenticação de canal para obter mais informações.
O ID do usuário que é definido no atributo **MCAUSER** na definição de canal SVRCONN	Usado a menos que substituído por uma saída de segurança ou uma regra CHLAUTH.
O ID do usuário que é transmitido a partir da máquina cliente	Usado quando nenhum ID do usuário é definido por qualquer outro meio.
O ID do usuário que iniciou o canal de conexão do servidor	Usado quando nenhum ID do usuário é configurado por qualquer outro meio e nenhum ID do usuário cliente é transmitido. Consulte a seção a seguir, "O ID do usuário que executa o programa do canal" na página 109 para obter mais informações.

Como o MCA de conexão do servidor faz chamadas de MQI em nome de usuários remotos, é importante considerar as implicações de segurança do MCA de conexão do servidor que emite chamadas de MQI em nome de clientes remotos, e como administrar o acesso de um número potencialmente grande de usuários.

- Uma abordagem é para a conexão do servidor MCA emitir chamadas MQI em sua própria autoridade. Mas tenha cuidado, normalmente é indesejável para a conexão do servidor MCA, com seus recursos de acesso poderosos, emitir chamadas MQI em nome de usuários clientes.
- Outra abordagem é usar o ID do usuário que flui a partir do cliente. A conexão do servidor MCA pode emitir chamadas MQI usando os recursos de acesso do ID do usuário do cliente. Esta abordagem apresenta várias questões a considerar:
  1. Existem diferentes formatos para o ID do usuário em diferentes plataformas. Isto às vezes causa problemas se o formato do ID do usuário no cliente diferir dos formatos aceitáveis no servidor.
  2. Há potencialmente muitos clientes com IDs de usuário diferentes e em mudança. Os IDs precisam ser definidos e gerenciados no servidor.
  3. O ID do usuário é confiável? Qualquer ID do usuário pode fluir a partir de um cliente, não necessariamente o ID do usuário que efetuou logon. Por exemplo, o cliente pode fluir um ID com total autoridade `mqm` que foi intencionalmente definida apenas no servidor por razões de segurança.
- A abordagem preferencial é definir tokens de identificação de cliente no servidor e, portanto, limitar os recursos de aplicativos conectados pelo cliente. Isto é geralmente feito configurando a propriedade MCAUSER do canal de conexão do servidor com um valor de ID do usuário especial a ser usado pelos clientes, e definindo alguns IDs para uso por clientes com nível diferente de autorização no servidor.

Configurando o ID do usuário em uma saída de segurança

Para o IBM MQ MQI clients, o processo que emite as chamadas de MQI é o MCA de conexão do servidor MCA. O ID do usuário usado pela conexão do servidor MCA está contido nos campos `MCAUserIdentifier` ou `LongMCAUserIdentifier` do MQCD. O conteúdo destes campos é configurado por:

- Qualquer valor configurado pelas saídas de segurança
- O ID do usuário do cliente
- MCAUSER (na definição do canal de conexão do servidor)

A saída de segurança pode substituir os valores que estão visíveis para ela, quando ela é invocada.

- Se o atributo MCAUSER do canal de conexão do servidor estiver configurado como não-em branco, o valor MCAUSER será usado.
- Se o atributo MCAUSER do canal de conexão do servidor estiver em branco, o ID do usuário recebido do cliente será usado.
- Se o atributo MCAUSER do canal de conexão do servidor estiver em branco, e nenhum ID do usuário for recebido do cliente, o ID do usuário que iniciou o canal de conexão do servidor será usado.

O cliente do IBM MQ não flui o ID do usuário declarado para o servidor quando uma saída de segurança do lado do cliente está em uso.

O ID do usuário que executa o programa do canal

Quando os campos de ID do usuário forem derivados do ID do usuário que iniciou o canal de conexão do servidor, o seguinte valor será usado:

- z/OS Para o z/OS, o ID do usuário designado à tarefa iniciada pelo inicializador de canais pela tabela de procedimentos iniciados do z/OS.
- Para TCP/IP (não z/OS), o ID do usuário da entrada `inetd.conf` ou o ID do usuário que iniciou o listener.
- Para SNA (não z/OS), o ID do usuário da entrada do Servidor SNA ou (se não houver nenhum) o pedido de conexão recebido ou o ID do usuário que iniciou o listener.
- Para o NetBIOS ou SPX, o ID do usuário que iniciou o listener.

Se qualquer definição de canal de conexão do servidor existir tendo o atributo MCAUSER configurado como em branco, os clientes poderão usar esta definição de canal para se conectar ao gerenciador de filas com autoridade de acesso determinada pelo ID do usuário fornecido pelo cliente. Pode haver uma exposição de segurança se o sistema em que o gerenciador de filas está sendo executado permitir conexões de rede não-autorizadas. O canal de conexão do servidor padrão IBM MQ (SYSTEM.DEF.SVRCONN) possui o atributo MCAUSER configurado para em branco. Para evitar acesso não autorizado, atualize o atributo MCAUSER da definição padrão com um ID do usuário que não possui acesso aos objetos do IBM MQ MQ.

Caso de IDs de usuário

Quando você define um canal com `runmqsc`, o atributo MCAUSER é alterado para letra maiúscula a menos que o ID do usuário esteja contido entre aspas simples.

ALW Para servidores no AIX, Linux, and Windows, o conteúdo do campo `MCAUserIdentifier` que é recebido do cliente muda para minúsculo.

IBM i Para servidores no IBM i, o conteúdo do campo `LongMCAUserIdentifier` que é recebido do cliente é mudado para letras maiúsculas.

Linux AIX Para servidores nos sistemas AIX and Linux, o conteúdo do campo `LongMCAUserIdentifier` que é recebido do cliente é mudado para letras minúsculas.

Por padrão, o ID do usuário, que é transmitido quando um aplicativo de ligação IBM MQ JMS é usado, é o ID do usuário para a JVM no qual o aplicativo está em execução.

Também é possível transmitir um ID de usuário por meio do método `createQueueConnection`.

Planejando a confidencialidade

Planeje como manter seus dados confidenciais.

É possível implementar confidencialidade no nível do aplicativo ou no nível de link. É possível escolher usar TLS nos casos em que se deve planejar o uso de certificados digitais. Também é possível usar programas de saída de canal se os recursos padrão não atenderem aos seus requisitos.

Conceitos relacionados

"Comparando a segurança no nível do link com a segurança no nível do aplicativo" na página 110
Este tópico contém informações sobre os diversos aspectos da segurança em nível de aplicativo e segurança em nível de link, e compara os dois níveis de segurança.

"Programas de Saída de Canal" na página 115
Os *programas de saída de Canal* são programas chamados em lugares definidos na seqüência do processamento de um MCA. Usuários e fornecedores podem desenvolver seus próprios programas de saída de canal. Alguns são fornecidos pelo IBM.

"Protegendo canais com SSL/TLS" na página 122
O suporte de TLS no IBM MQ usa o objeto de informações sobre autenticação do gerenciador de filas e vários comandos MQSC. Também se deve considerar o uso de certificados digitais.

Comparando a segurança no nível do link com a segurança no nível do aplicativo

Este tópico contém informações sobre os diversos aspectos da segurança em nível de aplicativo e segurança em nível de link, e compara os dois níveis de segurança.

A segurança em nível de link e de aplicativo é ilustrada na Figura 10 na página 110.

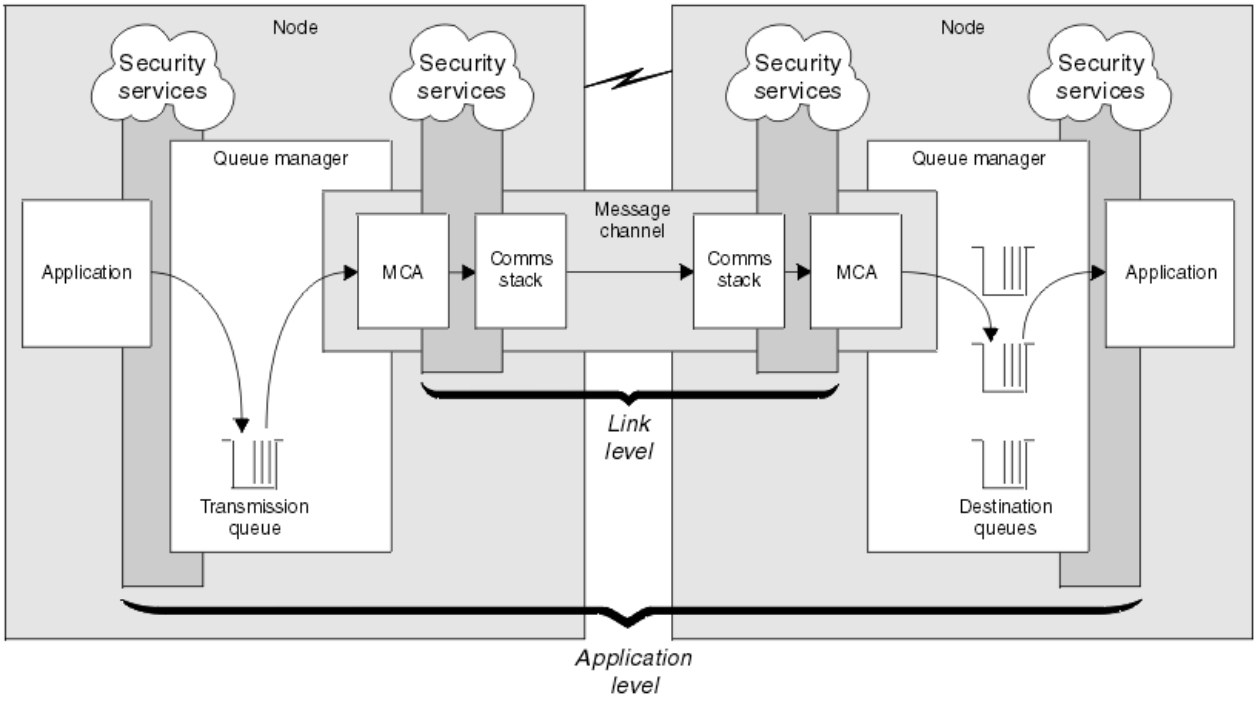


Figura 10. Segurança no nível do link e segurança no nível do aplicativo

Protegendo mensagens em filas

A segurança no nível do link pode proteger mensagens enquanto são transferidas de um gerenciador de filas para outro. É de particular importância quando as mensagens são transmitidas através de uma rede não protegida. Ela não pode, porém, proteger as mensagens enquanto elas estão armazenadas em filas em um gerenciador de filas de origem, de destino ou intermediário.

z/OS A criptografia do conjunto de dados do z/OS pode fornecer alguma proteção de mensagens armazenadas em filas, mas somente para dados em repouso em um gerenciador de filas locais. Consulte a seção Confidencialidade para dados em repouso no IBM MQ for z/OS com criptografia do conjunto de dados. para obter informações adicionais.

A segurança em nível de aplicativo, em comparação, pode proteger as mensagens enquanto elas estão armazenadas em filas e se aplica mesmo quando não está sendo utilizado enfileiramento distribuído. Essa é a principal diferença entre a segurança no nível do link e a segurança no nível do aplicativo, e é ilustrada na Figura 10 na página 110.

Gerenciadores de filas que não estão executando em ambientes controlados e de confiança

Se um gerenciador de filas estiver em execução em um ambiente controlado e confiável, os mecanismos de controle de acesso fornecidos pelo IBM MQ podem ser considerados suficientes para proteger as mensagens armazenadas em suas filas. Isso é especialmente verdadeiro se somente filas locais estiverem envolvidas e as mensagens nunca deixarem o gerenciador de filas. A segurança no nível do aplicativo neste caso pode ser considerada desnecessária.

Ela também pode ser considerada desnecessária se as mensagens forem transferidas para outro gerenciador de filas que também esteja executando em um ambiente controlado e de confiança, ou forem recebidas de um gerenciador de filas nessas condições. A necessidade de segurança em nível de aplicativo torna-se maior quando as mensagens são transferidas ou recebidas de um gerenciador de filas que não está sendo executado em um ambiente controlado e confiável.

Diferenças de custo

A segurança no nível do aplicativo pode custar mais que a segurança no nível do link em termos de administração e de desempenho.

É provável que o custo da administração seja maior porque existem mais restrições para configurar e manter. Por exemplo, você pode precisar assegurar que um determinado usuário envia somente certos tipos de mensagens e envia mensagens somente para certos destinos. Por outro lado, você pode precisar assegurar que um determinado usuário recebe somente certos tipos de mensagens e recebe mensagens somente de certas origens. Em vez de gerenciar os serviços de segurança no nível do link em um único canal de mensagem, você pode precisar configurar e manter regras para cada par de usuários que trocam mensagens através desse canal.

Pode haver um impacto no desempenho se os serviços de segurança forem chamados sempre que um aplicativo colocar ou obtiver uma mensagem.

As organizações tendem a considerar a segurança no nível do link primeiro porque ela pode ser mais fácil de implementar. Elas consideram a segurança em nível de aplicativo se descobrem que a segurança no nível do link não satisfaz todos os seus requisitos.

Disponibilidade de componentes

Geralmente, em um ambiente distribuído, um serviço de segurança requer um componente em pelo menos dois sistemas. Por exemplo, uma mensagem pode ser criptografada em um sistema e descriptografada em outro. Isso se aplica tanto à segurança no nível do link quanto à segurança em nível de aplicativo.

Em um ambiente heterogêneo, com diferentes plataformas em uso, cada uma delas com deferentes níveis de funções de segurança, os componentes necessários de um serviço de segurança podem não

estar disponíveis para todas as plataformas nas quais eles são necessários e de uma forma fácil de utilizar. Isso provavelmente é um problema maior para a segurança em nível de aplicativo que para a segurança no nível do link, especialmente se você pretender fornecer sua própria segurança em nível de aplicativo comprando componentes de várias origens.

Mensagens em uma fila de cartas não entregues

Se uma mensagem for protegida pela segurança em nível de aplicativo, pode haver um problema se, por algum motivo, a mensagem não atingir seu destino e for colocada em uma fila de cartas não entregues. Se você não descobrir como processar a mensagem a partir das informações no descritor da mensagem e do cabeçalho da carta não entregue, poderá precisar inspecionar o conteúdo dos dados do aplicativo. Não é possível fazer isso se os dados do aplicativo estiverem criptografados e somente o destinatário pretendido poderá decriptografá-los.

O que a segurança em nível de aplicativo não pode fazer

A segurança no nível do aplicativo não é uma solução completa. Mesmo se você implementar a segurança em nível de aplicativo, alguns serviços da segurança no nível do link ainda podem ser necessários. Por exemplo:

- Quando um canal inicia, a autenticação mútua dos dois MCAs pode ainda ser uma exigência. Isso pode ser feito somente por um serviço de segurança no nível do link.
- A segurança em nível de aplicativo não pode proteger o cabeçalho da fila de transmissão, MQXQH, o qual inclui o descritor da mensagem incorporado. Também não se pode proteger os dados nos fluxos de protocolo do canal do IBM MQ diferentes dos dados da mensagem. Somente a segurança no nível do link pode fornecer essa proteção.
- Se os serviços da segurança em nível de aplicativo forem chamados na extremidade do servidor de um canal MQI, os serviços não poderão proteger os parâmetros das chamadas de MQI que forem enviadas pelo canal. Em particular, os dados do aplicativo em um chamada MQPUT, MQPUT1 ou MQGET não são protegidos. Somente a segurança no nível do link pode fornecer a proteção neste caso.

Segurança no nível do link

Segurança no nível do link se refere ao serviços de segurança que são chamados, direta ou indiretamente, por um MCA, pelo subsistema de comunicações ou por uma combinação dos dois trabalhando em conjunto.

A segurança em nível de link é ilustrada no Figura 10 na página 110.

Eis alguns exemplos de serviços de segurança no nível do link:

- O MCA em cada extremidade de um canal de mensagem pode autenticar seu parceiro. Isso é feito quando o canal iniciar e uma conexão de comunicações tiver sido estabelecido, mas antes que as mensagens comecem a fluir. Se a autenticação falhar em qualquer das extremidades, o canal será fechado e nenhuma mensagem será transferida. Este é um exemplo de um serviço de identificação e autenticação.
- Uma mensagem pode ser criptografada na extremidade de envio de um canal e decriptografada na extremidade de recepção. Este é um exemplo de um serviço de confidencialidade.
- Uma mensagem pode ser verificada na extremidade de recepção de um canal para determinar se seu conteúdo foi modificado deliberadamente enquanto ela estava sendo transmitida pela rede. Este é um exemplo de um serviço de integridade de dados.

A segurança em nível de link fornecida pelo IBM MQ

O meio primário de provisão de confidencialidade e integridade de dados no IBM MQ é pelo uso de TLS. Para obter mais informações sobre o uso de TLS no IBM MQ, veja “Protocolos de segurança TLS no IBM MQ” na página 25. Para autenticação, o IBM MQ fornece o recurso para usar registros de autenticação de canal. Os registros de autenticação de canal oferecem controle preciso sobre o acesso concedido à conexão de sistemas, no nível de canais individuais ou grupos de canais. Para obter mais informações, consulte “Registros de Autenticação de Canal” na página 53.

Fornecendo sua Própria Segurança em Nível de Link
É possível fornecer seus próprios serviços de segurança de nível de link. Gravar seus próprios programas de saída do canal é a principal forma de fornecer seus próprios serviços de segurança em nível de link.

Os programas de saída de canal são apresentados em "Programas de Saída de Canal" na página 115. O mesmo tópico também descreve o programa de saída do canal que é fornecido com o IBM MQ for Windows (o programa de saída do canal SSPI). Este programa de saída de canal é fornecido em formato de fonte para que você possa alterar o código-fonte para se adequar a suas necessidades. Se este programa de saída do canal, ou programas de saída do canal disponíveis a partir de outros fornecedores, não atenderem seus requisitos, será possível projetar e gravar seu próprio. Este tópico sugere maneiras nas quais os programas de saída do canal podem fornecer serviços de segurança. Para obter informações sobre como gravar um programa de saída do canal, consulte o Gravando programas de saída do canal.

Segurança em Nível de Link Usando uma Saída de Segurança
As saídas de segurança normalmente trabalham em pares; uma em cada extremidade de um canal. Elas são chamadas imediatamente após a conclusão da negociação de dados inicial na inicialização do canal.

Saídas de segurança podem ser usadas para fornecer identificação e autenticação, controle de acesso e confidencialidade.

Segurança em Nível de Link Usando uma Saída de Mensagem
Uma saída de mensagem pode ser utilizada apenas em um canal de mensagem, não em um canal MQI. Ela tem acesso ao cabeçalho da fila de transmissão, MQXQH, que inclui o descritor de mensagens internas e os dados do aplicativo em uma mensagem. Pode modificar o conteúdo da mensagem e alterar seu comprimento.

Uma saída de mensagem pode ser utilizada para qualquer objetivo que exija acesso à mensagem inteira, em vez de uma parte dela.

Saídas de mensagem podem ser usadas para fornecer identificação e autenticação, controle de acesso, confidencialidade, integridade de dados e irrecusabilidade, e por outros motivos que não segurança.

Segurança em Nível de Link Usando Saídas de Envio e Recebimento
As saídas de envio e recebimento podem ser utilizadas nos canais de mensagem e MQI. Podem ser chamadas para todos os tipos de dados que passam em um canal e para fluxos nas duas direções.

As saídas de envio e recebimento têm acesso a cada segmento de transmissão. Elas podem modificar seu conteúdo e alterar seu comprimento.

Em um canal de mensagens, se um MCA tem que dividir uma mensagem e enviá-la em mais de um segmento de transmissão, uma saída de envio poderá ser chamada para cada segmento de transmissão que contém uma parte da mensagem e, na extremidade de recebimento, uma saída de recebimento é chamada para cada segmento de transmissão. O mesmo ocorrerá em um canal MQI se os parâmetros de entrada ou saída de uma chamada MQI forem muito grandes para serem enviados em um único segmento de transmissão.

Em um canal MQI, o byte 10 de um segmento de transmissão identifica a chamada MQI e indica se o segmento de transmissão contém os parâmetros de entrada ou saída da chamada. As saídas de envio e recebimento podem examinar este byte para determinar se a chamada MQI contém dados do aplicativo que podem necessitar de proteção.

Quando uma saída de usuário é chamada pela primeira vez, para adquirir e inicializar os recursos de que necessita, pode solicitar que o MCA reserve uma quantidade específica de espaço no buffer que contém um segmento de transmissão. Quando é chamada posteriormente para processar um segmento de transmissão, ela pode usar esse espaço para incluir uma chave criptografada ou uma assinatura digital, por exemplo. A saída de recepção correspondente na outra extremidade do canal pode remover os dados incluídos pela saída de envio e utilizá-los para processar o segmento de transmissão.

As saídas de envio e recebimento são mais adequadas para propósitos em que não precisam entender a estrutura dos dados que estão manipulando, podendo, assim, tratar cada segmento de transmissão como um objeto binário.

As saídas de envio e recebimento podem ser usadas para fornecer confidencialidade e integridade de dados, e para outros usos que não segurança.

Tarefas relacionadas

Identificando uma chamada de API em um programa de saída de envio ou recebimento

Segurança em Nível de Aplicativo

Segurança no nível do aplicativo se refere aos serviços de segurança que são chamados na interface entre um aplicativo e um gerenciador de filas ao qual ele está conectado.

Esses serviços são chamados quando o aplicativo emite chamadas de MQI para o gerenciador de filas. Os serviços podem ser chamados direta ou indiretamente, pelo aplicativo, pelo gerenciador de filas, por outro produto que suporta o IBM MQ ou por uma combinação de qualquer um desses trabalhando em conjunto. A segurança em nível de aplicativo é ilustrada na Figura 10 na página 110.

A segurança em nível de aplicativo também é conhecida como *segurança ponta-a-ponta* ou *segurança no nível da mensagem*.

Eis alguns exemplos de serviços de segurança em nível de aplicativo:

- quando um aplicativo coloca uma mensagem em uma fila, o descritor da mensagem contém um ID de usuário associado ao aplicativo. Entretanto, não existem dados presentes, tais como uma senha criptografada, que possam ser utilizados para autenticar o ID do usuário. Um serviço de segurança pode incluir esses dados. Quando a mensagem for finalmente recuperada pelo aplicativo de recepção, outro componente do serviço pode autenticar o ID do usuário utilizando os dados que foram enviados com a mensagem. Este é um exemplo de um serviço de identificação e autenticação.
- Uma mensagem pode ser criptografada quando é colocada em uma fila por um aplicativo, e descriptografada quando é recuperada pelo aplicativo de recepção.Este é um exemplo de um serviço de confidencialidade.
- Uma mensagem pode ser verificada quando é recuperada pelo aplicativo de recepção. Essa verificação determina se seu conteúdo foi modificado deliberadamente desde que foi colocada pela primeira vez em uma fila pelo aplicativo de envio.Este é um exemplo de um serviço de integridade de dados.

Planejamento para o Advanced Message Security

O Advanced Message Security (AMS) é um componente do IBM MQ que fornece um alto nível de proteção para dados sensíveis que fluem por meio da rede do IBM MQ, ao mesmo tempo em que não impacta os aplicativos finais.

Se você estiver movendo informações altamente sigilosas ou de valor, informações especialmente confidenciais ou relacionadas a pagamento, como registros de paciente ou detalhes de cartão de crédito, deverá prestar muita atenção à segurança de informações. Assegurar que as informações que passam pela empresa retenham sua integridade e sejam protegidas contra acesso não autorizado é um desafio e uma responsabilidade contínua. É provável também que você seja obrigado a cumprir os regulamentos de segurança, sob o risco de penalidades por falta de conformidade.

É possível desenvolver suas próprias extensões de segurança para o IBM MQ. No entanto, essas soluções requerem qualificações de especialistas e podem ser complicadas e caras de se manter. O Advanced Message Security ajuda a lidar com esses desafios ao mover informações acerca da empresa entre cada tipo de sistema de TI comercial virtualmente.

O Advanced Message Security estende os recursos de segurança do IBM MQ das seguintes maneiras:

- Fornece proteção de dados de ponta a ponta no nível do aplicativo para sua infraestrutura do sistema de mensagens ponto a ponto, usando criptografia ou assinatura digital de mensagens.
- Fornece segurança abrangente sem gravar código de segurança complexo ou modificar ou recompilar aplicativos existentes.
- Usa a tecnologia de Infraestrutura de Chave Pública (PKI) para fornecer autenticação, autorização, confidencialidade e serviços de integridade de dados para mensagens.
- Fornece administração de políticas de segurança para servidores mainframe e distribuídos.
- Ele suporta servidores e clientes do IBM MQ.

- Ele se integra com o Managed File Transfer para fornecer uma solução do sistema de mensagens segura de ponta a ponta.

Para obter mais informações, consulte “Advanced Message Security” na página 608.

Fornecendo sua própria segurança de nível de aplicativo
É possível fornecer seus próprios serviços de segurança de nível de aplicativo. Para ajudar a implementar a segurança no nível do aplicativo, o IBM MQ fornece duas saídas, a saída da API e a saída cruzada da API.

A saída de API e a saída cruzada da API podem fornecer identificação e autenticação, controle de acesso, confidencialidade, integridade de dados, serviços de não repúdio e outras funções não relacionadas à segurança.

Se a saída API ou a saída de cruzamento de API não for suportada em seu ambiente de sistema, você pode desejar considerar outras maneiras de fornecer sua própria segurança de nível de aplicativo. Uma maneira é desenvolver uma API de nível mais alto que encapsule a MQI. Os programadores usam, então, essa API, em vez do MQI, para gravar aplicativos do IBM MQ.

As razões mais comuns para utilizar uma API de nível mais alto são:

- Para ocultar os recursos mais avançados da MQI dos programadores.
- Para reforçar padrões na utilização da MQI.
- Para incluir uma função à MQI. Esta função adicional pode ser serviços de segurança.

Alguns produtos de fornecedores usam esta técnica para fornecer segurança de nível do aplicativo para IBM MQ.

Se você estiver planejando fornecer serviços de segurança desta maneira, observe o seguinte em relação à conversão de dados:

- Se um token de segurança, tal como uma assinatura digital, tiver sido incluído aos dados de aplicativo na mensagem, qualquer código executando conversão de dados deve estar ciente da presença deste token.
- Um token de segurança pode ser derivado de uma imagem binária dos dados do aplicativo. Portanto, qualquer verificação do token deve ser feita antes de converter os dados.
- Se os dados do aplicativo na mensagem tiverem sido criptografados, eles devem ser decriptografados antes da conversão de dados.

Programas de Saída de Canal

Os *programas de saída de Canal* são programas chamados em lugares definidos na seqüência do processamento de um MCA. Usuários e fornecedores podem desenvolver seus próprios programas de saída de canal. Alguns são fornecidos pelo IBM.

Existem diversos tipos de programas de saída de canal, mas apenas quatro têm uma função de fornecer segurança em nível de link:

- Saída de segurança
- Saída de mensagen
- Saída de envio
- Saída de recepção

Esses quatro tipos de programas de saída de canal são ilustrados na Figura 11 na página 116 e descritos nos tópicos a seguir.

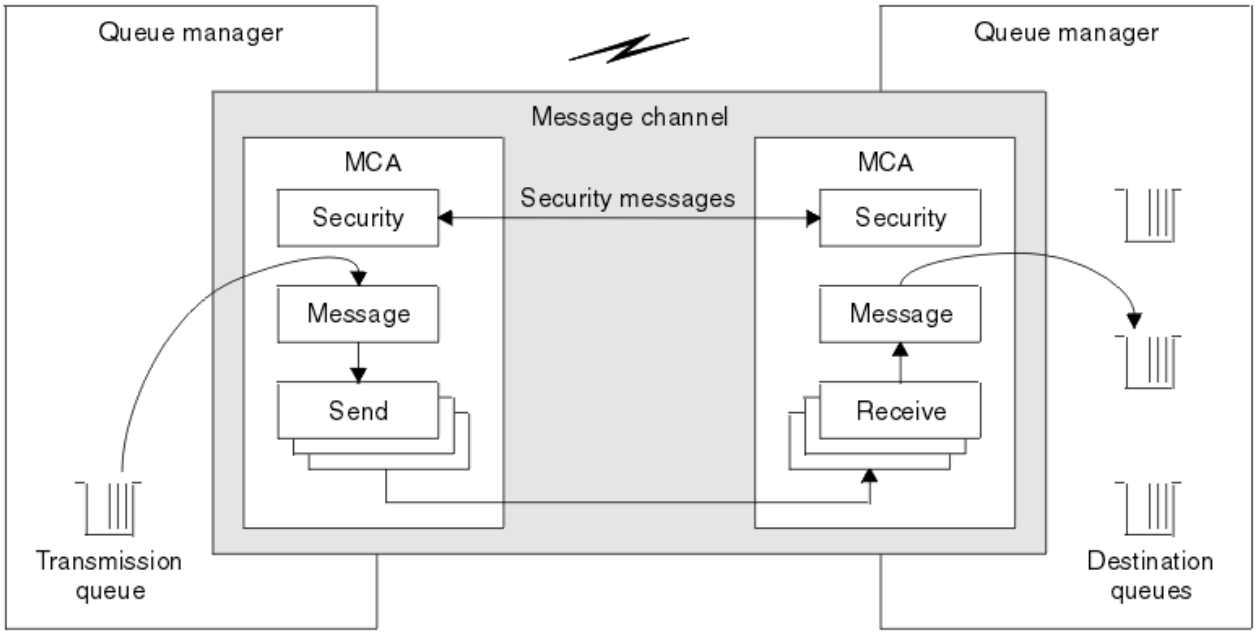


Figura 11. Saídas de segurança, mensagem, envio e recebimento em um canal de mensagens

Conceitos relacionados

Programas de Saída de Canal para Canais de Mensagens

Visão Geral da Saída de Segurança

As saídas de segurança normalmente trabalham em pares. Elas são chamadas antes dos fluxos de mensagens e seus propósitos são permitir que um MCA autentique seu parceiro.

As *Saídas de segurança* normalmente trabalham em pares; uma em cada extremidade de um canal. Elas são chamadas imediatamente após a conclusão da negociação de dados inicial na inicialização do canal, mas antes do início do fluxo de qualquer mensagem. A principal finalidade da saída de segurança é permitir ao MCA de cada extremidade de um canal autenticar o seu parceiro. No entanto, não há nada que possa evitar que uma saída de segurança efetue outra função, mesmo uma função que não tenha nada a ver com segurança.

As saídas de segurança podem se comunicar umas com as outras enviando *mensagens de segurança*. O formato de uma mensagem de segurança não é definido e é determinado pelo usuário. Um possível resultado da troca de mensagens de segurança é que uma das saídas de segurança pode decidir não prosseguir mais. Nesse caso, o canal é fechado e as mensagens não fluem. Se existir uma saída de segurança em apenas uma extremidade de um canal, a saída ainda é chamada e pode escolher se vai prosseguir ou fechar o canal.

As saídas de segurança podem ser chamadas em canais de mensagens e do MQI. O nome de uma saída de segurança é especificado como um parâmetro na definição do canal em cada extremidade de um canal.

Para obter mais informações sobre saídas de segurança, consulte “Segurança em Nível de Link Usando uma Saída de Segurança” na página 113.

Saída de mensagen

As saídas de mensagens operam apenas em canais de mensagens e normalmente funcionam em pares. Uma saída de mensagem pode operar em toda a mensagem e fazer várias mudanças nela.

As *Saídas de mensagens* nas extremidades de envio e de recebimento de um canal normalmente trabalham em pares. Uma saída de mensagens na extremidade de envio de um canal é chamada após o MCA ter recebido uma mensagem da fila de transmissão. Na extremidade de recebimento de um canal, uma saída de mensagens é chamada antes que o MCA coloque uma mensagem em sua fila de destino.

Uma saída de mensagens tem acesso ao cabeçalho da fila de transmissão, MQXQH, que inclui o descritor de mensagens embutidas, e os dados do aplicativo em uma mensagem. Uma saída de mensagens pode modificar o conteúdo da mensagem e alterar seu comprimento. Uma alteração no comprimento pode ser o resultado da compressão, descompressão, criptografia e decriptografia da mensagem. Também pode ser o resultado de incluir dados na mensagem ou remover dados dela.

As saídas de mensagens podem ser utilizadas para qualquer finalidade que exija acesso à mensagem inteira, em vez de parte dela, e não necessariamente para segurança.

Uma saída de mensagem pode determinar que a mensagem que está processando atualmente não deve continuar além da direção de seu destino. O MCA coloca a mensagem na fila dead-letter. Uma saída de mensagem pode também fechar o canal.

As saídas de mensagens podem ser chamadas apenas em canais de mensagens, não em canais do MQI. Isto ocorre porque o objetivo de um canal do MQI é permitir o fluxo de parâmetros de entrada e de saída de chamadas do MQI entre o aplicativo IBM MQ MQI client e o gerenciador de filas.

O nome de uma saída de mensagens é especificado como um parâmetro na definição do canal em cada extremidade de um canal. Você também pode especificar uma lista de saídas de mensagens para serem executadas sucessivamente.

Para obter mais informações sobre saídas de mensagem, consulte “Segurança em Nível de Link Usando uma Saída de Mensagem” na página 113.

Saídas de Envio e Recebimento

As saídas de envio e recebimento geralmente trabalham em pares. Elas operam em segmentos de transmissão e são melhor usadas onde a estrutura dos dados que estão processando não for relevante.

Uma *saída de envio* em uma extremidade de um canal e uma *saída de recebimento* na outra extremidade normalmente trabalham em pares. Uma saída de envio é chamada pouco antes de um MCA emitir um envio de comunicação para enviar dados para uma conexão de comunicação. Uma saída de recebimento é chamada logo depois que um MCA recuperou o controle após um recebimento de comunicação e recebeu dados de uma conexão de comunicação. Se as conversações compartilhadas estiverem em uso sobre um canal MQI, uma instância diferente de uma saída de envio e de recebimento será chamada para cada conversação.

Os fluxos de protocolo de canais do IBM MQ entre dois MCAs em um canal de mensagens contêm informações de controle, assim como dados das mensagens. Da mesma forma, em um canal do MQI, os fluxos contêm informações sobre controle assim como os parâmetros das chamadas do MQI. As saídas de envio e recebimento são chamadas para todos os tipos de dados.

Os dados de mensagens fluem em apenas uma direção em um canal de mensagens mas, em um canal do MQI, os parâmetros de entrada de uma chamada do MQI fluem em uma direção e os parâmetros de saída fluem na outra direção. Em canais de mensagens e do MQI, as informações de controle fluem em ambas as direções. Como resultado, as saídas de envio e recebimento podem ser chamadas em ambas as extremidades de um canal.

A unidade de dados que é transmitida em um único fluxo entre dois MCAs é denominada um *segmento de transmissão*. As saídas de envio e recebimento têm acesso a cada segmento de transmissão. Elas podem modificar seu conteúdo e alterar seu comprimento. No entanto, uma saída de envio não deve alterar os oito primeiros bytes de um segmento de transmissão. Esses 8 bytes fazem parte do cabeçalho do protocolo do canal do IBM MQ. Também existem restrições em relação a quanto uma saída de envio pode aumentar o comprimento de um segmento de transmissão. Especificamente, uma saída de envio não pode aumentar seu comprimento além do máximo negociado entre os dois MCAs na inicialização do canal.

Em um canal de mensagens, se uma mensagem for muito grande para ser enviada em um único segmento de transmissão, o MCA de envio divide a mensagem e a envia em mais de um segmento de transmissão. Como conseqüência, uma saída de envio é chamada para cada segmento de transmissão contendo uma parte da mensagem e, na extremidade de recebimento, uma saída de recebimento é chamada para cada segmento de transmissão. O MCA de recebimento reconstitui a mensagem a partir dos segmentos de transmissão após serem processados pela saída de recebimento.

Da mesma forma, em um canal do MQI, os parâmetros de entrada ou de saída de uma chamada do MQI são enviados em mais de um segmento de transmissão se forem muito grandes. Isso pode ocorrer, por exemplo, em uma chamada MQPUT, MQPUT1 ou MQGET se os dados do aplicativo forem grandes o suficiente.

Levando em conta essas considerações, é mais apropriado utilizar saídas de envio e recebimento para objetivos nos quais elas não precisem entender a estrutura dos dados que estão tratando e possam, assim, tratar cada segmento de transmissão como um objeto binário.

Uma saída de envio ou de recebimento pode fechar um canal.

Os nomes de uma saída de envio e de uma saída de recebimento são especificados como parâmetros na definição do canal em cada extremidade de um canal. Você também pode especificar uma lista de saídas de envio a serem executadas sucessivamente. Da mesma maneira, você pode especificar uma lista de saídas de recebimento.

Para obter mais informações sobre saídas de envio e recebimento, consulte “Segurança em Nível de Link Usando Saídas de Envio e Recebimento” na página 113.

Planejando a integridade de dados

Planeje como preservar a integridade dos dados.

É possível implementar a integridade dos dados no nível do aplicativo ou no nível de link.

No nível do aplicativo, é possível usar programas de saída de API se os recursos padrão não satisfizerem seus requisitos. É possível optar por usar o Advanced Message Security (AMS) para assinar mensagens digitalmente para proteção contra modificação não autorizada.

No nível de link, você pode escolher usar TLS. Nesse caso, deve-se planejar o uso de certificados digitais. Também é possível usar programas de saída de canal se os recursos padrão não atenderem aos seus requisitos.

Conceitos relacionados

“Protegendo canais com SSL/TLS” na página 122
O suporte de TLS no IBM MQ usa o objeto de informações sobre autenticação do gerenciador de filas e vários comandos MQSC. Também se deve considerar o uso de certificados digitais.

“Integridade de dados” na página 10
O serviços *integridade dos dados* detecta se houve modificação não-autorizada dos dados.

“Planejamento para o Advanced Message Security” na página 114
O Advanced Message Security (AMS) é um componente do IBM MQ que fornece um alto nível de proteção para dados sensíveis que fluem por meio da rede do IBM MQ, ao mesmo tempo em que não impacta os aplicativos finais.

Referências relacionadas

Referência de saída de API

Chamadas de Saída do Canal e Estrutura de Dados

Planejando a auditoria

Decida quais dados são necessários para a auditoria e como as informações de auditoria serão capturadas e processadas. Considere como verificar se o sistema está configurado corretamente.

Há vários aspectos para o monitoramento de atividade. Os aspectos que devem ser considerados são frequentemente definidos por requisitos de auditoria, e esses requisitos são geralmente orientados por normas regulamentares, como o HIPAA (Health Insurance Portability and Accountability Act) ou o SOX (Sarbanes-Oxley). O IBM MQ fornece recursos destinados a ajudar na conformidade com tais normas.

Considere se você está interessado apenas em exceções ou se está interessado em todo o comportamento do sistema.

Alguns aspectos de auditoria também podem ser considerados como monitoramento operacional; uma distinção para a auditoria é que você está sempre olhando para dados históricos, e não apenas para alertas em tempo real. O monitoramento é abrangido na seção Monitoramento e desempenho.

Quais dados auditar

Considere quais tipos de dados ou atividade são necessários para auditar, conforme descrito nas seções a seguir:

As mudanças feitas no IBM MQ usando as interfaces do IBM MQ
Configure o IBM MQ para emitir os eventos de instrumentação, especificamente os eventos de comando e eventos de configuração.

Mudanças feitas no IBM MQ fora de seu controle
Algumas mudanças podem afetar como o IBM MQ se comporta, mas não podem ser monitoradas diretamente pelo IBM MQ. Exemplos de tais mudanças incluem mudanças nos arquivos de configuração `mqs.ini`, `qm.ini` e `mqclient.ini`, a criação e a exclusão de gerenciadores de filas, instalação de arquivos binários como programas de saída de usuários e alterações de permissões de arquivo. Para monitorar essas atividades, deve-se usar ferramentas em execução no nível do sistema operacional. Diferentes ferramentas estão disponíveis e são apropriadas para diferentes sistemas operacionais. Também é necessário ter logs criados por ferramentas associadas, tais como *sudo*.

Controle operacional do IBM MQ
Talvez seja necessário usar as ferramentas do sistema operacional para auditar atividades, como iniciar e parar os gerenciadores de filas. Em alguns casos, IBM MQ pode ser configurado para emitir eventos de instrumentação.

Atividade do aplicativo no IBM MQ
Para auditar as ações de aplicativos, por exemplo, abrir filas e enviar e receber mensagens, configure o IBM MQ para emitir eventos adequados.

Alertas de intruso
Para auditar tentativas de violação de segurança, configure o sistema para emitir eventos de autorização. Os eventos do canal também podem ser úteis para mostrar a atividade, especialmente se um canal for encerrado inesperadamente.

Planejando a captura, exibição e arquivamento de dados de auditoria

Muitos dos elementos que são necessários são relatados como mensagens do evento do IBM MQ. Deve-se escolher ferramentas que podem ser lidas e formatar essas mensagens. Se você estiver interessado em armazenamento e análise de longo prazo, deverá movê-los para um mecanismo de armazenamento auxiliar, como um banco de dados. Se essas mensagens não forem processadas, elas permanecerão na fila de eventos, possivelmente preenchendo a fila. É possível escolher implementar uma ferramenta que executa automaticamente uma ação com base em alguns eventos; por exemplo, para emitir um alerta quando uma falha de segurança ocorre.

Verificando se seu sistema está corretamente configurado

Um conjunto de testes é fornecido com o IBM MQ Explorer. Use estes testes para verificar problemas em suas definições de objeto.

Além disso, verifique periodicamente se a configuração do sistema está conforme o esperado. Embora eventos de comando e configuração possam relatar quando algo é mudado, eles também são úteis para fazer dump da configuração e compará-la com uma cópia correta conhecida.

Planejando a segurança por meio da topologia

Esta seção abrange a segurança em situações específicas, nomeadamente para os canais, clusters de gerenciadores de filas, publicação/assinatura e aplicativos multicast, e ao utilizar um firewall.

Consulte os subtópicos a seguir para obter mais informações:

Autorização de canal

Ao enviar ou receber uma mensagem por meio de um canal, será necessário fornecer acesso a diversos recursos do IBM MQ. Os agentes do canal de mensagens (MCAs) são essencialmente aplicativos do IBM MQ que movem mensagens entre gerenciadores de fila, e como tal, requerem acesso a vários recursos do IBM MQ para operar corretamente.

Para receber mensagens no tempo de PUT para MCAs, é possível usar o ID de usuário associado ao MCA ou o ID do usuário associado à mensagem.

No tempo CONNECT, é possível mapear o ID do usuário declarado para um usuário alternativo, usando registros de autenticação do canal **CHLAUTH**.

No IBM MQ, os canais podem ser protegidos pelo suporte de TLS.

Os IDs de usuário associados ao envio e recebimento de canais, excluindo o canal emissor em que o atributo MCAUSER não é usado, precisam ter acesso aos seguintes recursos:

- O ID do usuário associado a um canal de envio requer acesso ao gerenciador de filas, a fila de transmissão, a fila de mensagens não entregues e acesso a quaisquer outros recursos que são requeridos por saídas do canal.
- O ID do usuário MCAUSER de um canal receptor precisa da autoridade *+setall*. A razão é que o canal receptor tem que criar o MQMD completo, incluindo todos os campos de contexto, usando os dados que ele recebeu do canal emissor remoto. O gerenciador de filas, portanto, requer que o usuário que executa esta atividade tenha a autoridade *+setall*. Esta autoridade *+setall* deve ser concedida ao usuário para:
  - Todas as filas que o canal receptor coloca validamente as mensagens.
  - O objeto do gerenciador de filas. Para obter mais informações, veja Autorizações para contexto.
- O ID do usuário MCAUSER de um canal receptor no qual o originador solicitou uma mensagem de relatório COA precisa de autoridade *+passid* na fila de transmissão que retorna a mensagem de relatório. Sem essa autoridade, as mensagens de erro AMQ8077 são registradas.
- Com o ID do usuário associado ao canal de recebimento, é possível abrir as filas de destino para colocar mensagens nas filas. Isso envolve a Interface de enfileiramento de mensagens (MQI), portanto, as verificações de controle de acesso adicionais podem precisar ser feitas se você não estiver usando o gerenciador de autoridade de objeto (OAM) do IBM MQ. É possível especificar se as verificações de autorização são feitas em relação ao ID de usuário associado ao MCA (conforme descrito neste tópico) ou em relação ao ID de usuário associado à mensagem (a partir do campo MQMD UserIdentifier).

  Para os tipos de canais para os quais ele se aplica, o parâmetro **PUTAUT** de uma definição de canal especifica qual ID de usuário é usado para essas verificações.
  - O canal é padronizado para usar a conta de serviço do gerenciador de filas, que tem direitos administrativos integrais e não requer autorizações especiais.
  - No caso de canais de conexão do servidor, as conexões administrativas são bloqueadas por padrão pelas regras CHLAUTH e requerem fornecimento explícito.
  - Canais do tipo receptor, solicitante e receptor de cluster permitem administração local por qualquer gerenciador de filas adjacente, a menos que o administrador execute as etapas para restringir esse acesso.
  - Não é necessário conceder autoridade *dsp* e *ctrlx* para o ID do usuário MCAUSER de um canal receptor.
  - Antes da IBM MQ 8.0.0 Fix Pack 4, se você usa um ID do usuário que não possui os privilégios administrativos do IBM MQ, deve-se conceder autoridade **dsp** e **ctrlx** para o canal a esse ID do usuário para que o canal funcione.

    A partir da IBM MQ 8.0.0 Fix Pack 4, não há verificações de autoridade quando um canal ressincroniza a si mesmo e corrige os números de sequência.

    No entanto, a emissão de um comando RESET CHANNEL manualmente ainda requer **+dsp** e **+ctrlx** em todas as liberações.

Atenção: Quando a reconfiguração de um canal é necessária para confirmação do lote de mensagens, o IBM MQ tenta consultar o canal, o que requer autoridade **+dsp**.

- O atributo MCAUSER não é usado para o tipo de canal SDR.
- Se você usar o ID do usuário associado à mensagem, é provável que o ID do usuário seja de um sistema remoto. Esse ID do usuário do sistema remoto deve ser reconhecido pelo sistema de destino. Os comandos a seguir são exemplos do tipo de comando que é possível emitir para conceder autoridade a um ID do usuário de um sistema remoto:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect +inq +setall
```

```
setmqaut -m QMgrName -t chl -n Profile -g GroupName +dsp +ctrlx
```

em que *Profile* é um canal.

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

em que *Profile* é uma fila de mensagens não entregues, se configurada.

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

em que *Profile* é uma lista de filas autorizadas.

Atenção: Tome cuidado ao autorizar um ID do usuário a colocar mensagens na fila de comandos ou outras filas sensíveis do sistema.

O ID de usuário associado ao MCA depende do tipo de MCA. Há dois tipos de MCA:

MCA responsável pela chamada
MCAs que iniciam um canal. Os MCAs responsáveis pela chamada podem ser iniciados como processos individuais, como encadeamentos do inicializador de canais, ou como encadeamentos de um conjunto de processos. O ID do usuário usado é o ID do usuário associado ao processo pai (o inicializador de canais) ou o ID do usuário associado ao processo que inicia o MCA.

MCA respondente
MCAs respondentes são MCAs que são iniciados como resultado de uma solicitação feita por um MCA responsável pela chamada. MCAs respondentes podem ser iniciados como processos individuais, como encadeamentos do listener ou como encadeamentos de um conjunto de processos. O ID do usuário pode ser qualquer um dos tipos a seguir (nessa ordem, de preferência):

1. No APPC, o MCA responsável pela chamada pode indicar o ID de usuário a ser usado para o MCA respondente. Isso é chamado de ID do usuário da rede, e se aplica somente a canais iniciados como processos individuais. Configure o ID do usuário de rede usando o parâmetro USERID da definição de canal.
2. Se o parâmetro **USERID** não for usado, a definição de canal do MCA respondente pode especificar o ID do usuário que o MCA deve usar. Configure o ID do usuário usando o parâmetro **MCAUSER** da definição de canal.
3. Se o ID do usuário não foi definido por um dos métodos anteriores (dois), o ID do usuário do processo que inicia o MCA ou o ID do usuário do processo pai (o listener) é usado.

Conceitos relacionados

“Registros de Autenticação de Canal” na página 53
Para exercer um controle mais preciso sobre o acesso concedido à conexão de sistemas em um nível de canal, é possível usar registros de autenticação de canal.

Referências relacionadas

Propriedades de registro de autenticação de canal

Protegendo Definições do Inicializador de Canais

Apenas membros do grupo mqm podem manipular inicializadores de canais.

Os inicializadores de canais do IBM MQ não são objetos do IBM MQ; o acesso a eles não é controlado pelo OAM. O IBM MQ não permite que usuários ou aplicativos manipulem esses objetos, a menos que o ID do usuário seja um membro do grupo mqm. Se você tiver um aplicativo que emite o comando PCF **StartChannelInitiator**, o ID do usuário especificado no descritor de mensagens da mensagem PCF deve ser membro do grupo mqm no gerenciador de filas de destino.

Um ID de usuário também deverá ser um membro do grupo mqm na máquina de destino para emitir os comandos MQSC equivalentes por meio do comando PCF Escape ou usando `runmqsc` no modo indireto.

Filas de transmissão

Os gerenciadores de filas colocam mensagens remotas automaticamente em uma fila de transmissão; não é necessária nenhuma autoridade especial para isso.

No entanto, se for necessário colocar uma mensagem diretamente em uma fila de transmissão, isso exigirá autorização especial; consulte Tabela 12 na página 139.

Saídas do canal

Se registros de autenticação de canal não forem adequados, será possível usar saídas do canal para segurança incluída. Uma saída de segurança forma uma conexão segura entre dois programas de saída de segurança. Um programa é para o agente do canal de mensagens de envio (MCA) e o outro para o MCA de recebimento.

Consulte “Programas de Saída de Canal” na página 115 para obter mais informações sobre saídas do canal.

Protegendo canais com SSL/TLS

O suporte de TLS no IBM MQ usa o objeto de informações sobre autenticação do gerenciador de filas e vários comandos MQSC. Também se deve considerar o uso de certificados digitais.

Certificados digitais e repositórios de chaves

É uma boa prática configurar o atributo de rótulo do certificado do gerenciador de filas (**CERTLABL**) para o nome do certificado pessoal a ser usado para a maioria dos canais, e substituí-lo para exceções, configurando o rótulo certificado sobre os canais que requerem certificados diferentes.

Se são necessários muitos canais com certificados que diferem do certificado padrão definido no gerenciador de filas, deve-se considerar a divisão de canais entre os vários gerenciadores de filas ou usar um proxy MQIPT na frente do gerenciador de filas para apresentar um certificado diferente.

É possível usar um certificado diferente para cada canal, mas se você armazenar muitos certificados em um repositório de chaves, poderá esperar que o desempenho seja afetado ao iniciar os canais TLS. Tente manter o número de certificados em um repositório de chaves para menos de 50 e considere 100 como um máximo, pois o desempenho do IBM Global Security Kit (GSKit) diminui acentuadamente com repositórios de chaves maiores.

Permitir vários certificados no mesmo gerenciador de filas aumenta as chances de que vários certificados de autoridade de certificação sejam usados no mesmo gerenciador de filas. Isso aumenta as chances de o namespace Nome Distinto do Assunto do certificado entrar em conflito para certificados emitidos por autoridades de certificação separadas.

Enquanto autoridades de certificação profissionais são, provavelmente, mais cautelosas, autoridades de certificação internas muitas vezes não têm convenções de nomenclatura claras e você pode acabar com correspondências indesejadas entre uma CA e outra.

É necessário verificar o Nome Distinto do Emissor do certificado, além do Nome Distinto do Assunto. Para fazer isso, use um registro SSLPEERMAP de autenticação de canal e configure os campos **SSLPEER** e **SSLCERTI** para corresponder ao Nome Distinto do Assunto e Nome Distinto do Emissor respectivamente.

Certificados autoassinados e certificados assinados por CA

É importante planejar o uso de certificados digitais quando se está desenvolvendo e testando seu aplicativo, e para seu uso em produção. É possível usar os certificados assinados por CA ou os certificados autoassinados, dependendo do uso de seus gerenciadores de filas e aplicativos clientes.

Certificados assinados pelo CA
Para sistemas de produção, obtenha os certificados a partir de uma autoridade de certificação (CA) confiável. Ao se obter um certificado a partir de uma CA externa, você paga pelo serviço.

Certificados autoassinados
Enquanto você está desenvolvendo seu aplicativo, será possível usar certificados autoassinados ou certificados emitidos por uma autoridade de certificação local, dependendo da plataforma:

Em sistemas AIX, Linux, and Windows, é possível usar certificados autoassinados. Consulte a seção “Criando um certificado pessoal autoassinado no AIX, Linux, and Windows” na página 551 para obter instruções.

Em sistemas IBM i, é possível usar os certificados assinados pela autoridade de certificação local. Consulte a seção “Solicitando um certificado do servidor no IBM i” na página 290 para obter instruções.

No z/OS, é possível usar os certificados autoassinado ou assinado por CA local. Consulte “Creating a self-signed personal certificate on z/OS” na página 316 ou “Requesting a personal certificate on z/OS” na página 317 para obter instruções.

Os certificados autoassinados não são adequados para uso de produção, pelas seguintes razões:

- Certificados autoassinados não podem ser revogados, o que pode permitir que um invasor realize spoof em uma identidade após uma chave privada ter sido comprometida. CAs podem revogar um certificado comprometido, evitando seu uso adicional. O uso de certificados assinados por CA são, portanto, mais seguros em um ambiente de produção, embora certificados autoassinados sejam mais convenientes em um sistema de teste.
- Os certificados autoassinados nunca expirarão. Isso é conveniente e seguro em um ambiente de teste, mas em um ambiente de produção, isso os deixa abertos a eventuais violações de segurança. O risco é ainda composto pelo fato de os certificados autoassinados não poderem ser revogados.
- Um certificado autoassinado é usado como um certificado pessoal e como um certificado de autoridade de certificação raiz (ou âncora de confiança). Um usuário com um certificado pessoal autoassinado pode ser capaz de usá-lo para assinar outros certificados pessoais. Em geral, isso não é verdadeiro para certificados pessoais emitidos por uma autoridade de certificação, e representa uma exposição significativa.

CipherSpecs e certificados digitais

Somente um subconjunto dos CipherSpecs suportados pode ser usado com todos os tipos suportados de certificado digital. Portanto, é necessário escolher um CipherSpec apropriado para seus certificados digitais. Da mesma forma, se a política de segurança de sua organização requer que um determinado CipherSpec seja usado, deve-se obter os certificados digitais adequados.

Para obter mais informações sobre o relacionamento entre CipherSpecs e certificados digitais, consulte “Certificados digitais e compatibilidade de CipherSpec no IBM MQ” na página 48

Políticas de Validação de Certificado

O padrão IETF RFC 5280 especifica uma série de regras de validação de certificado que o software de aplicativo compatível deve implementar para impedir ataques de personificação. Um conjunto de regras de validação de certificado é conhecido como uma política de validação de certificado. Para obter mais informações sobre as políticas de validação de certificado no IBM MQ, consulte “Políticas de validação de certificado no IBM MQ” na página 46.

Planejando a verificação de revogação de certificado

Permitir vários certificados de autoridades de certificação diferentes causa, potencialmente, a verificação adicional de revogação de certificado desnecessária.

Em particular, se você tiver configurado explicitamente o uso de um servidor de revogação a partir de uma autoridade de certificação específica, por exemplo, usando um objeto AUTHINFO ou estrutura de registro de informações sobre autenticação (MQAIR), uma verificação de revogação falhará quando apresentada com um certificado a partir de uma autoridade de certificação diferente.

É necessário evitar a configuração do servidor de revogação de certificado explícito. Em vez disso, deve-se ativar a verificação implícita onde cada certificado contém seu próprio local de servidor de revogação em uma extensão de certificado, por exemplo, Ponto de Distribuição CRL ou OCSP AuthorityInfoAccess.

Para obter mais informações, consulte OCSPCheckExtensions e CDPCheckExtensions.

Comandos e atributos para o suporte de TLS

O protocolo Segurança da Camada de Transporte (TLS) fornece segurança de canal, com proteção contra espionagem do tráfego de rede, violação e personificação. O suporte do IBM MQ para TLS permite especificar, na definição de canal, que um determinado canal usa a segurança TLS. Também é possível especificar detalhes do tipo de segurança desejado, como o algoritmo de criptografia que você deseja usar.

- Os comandos do MQSC a seguir suportam TLS:

  ALTER AUTHINFO
  Modifica os atributos de um objeto de informações sobre autenticação.

  DEFINE AUTHINFO
  Cria um objeto de informações sobre autenticação.

  DELETE AUTHINFO
  Exclui um objeto de informações sobre autenticação.

  DISPLAY AUTHINFO
  Exibe os atributos para um objeto de informações sobre autenticação específico.

- Os parâmetros do gerenciador de filas a seguir suportam TLS:

  CERTLABL
  Define um rótulo de certificado pessoal a ser usado.

  KEYRPWD
  Em sistemas AIX, Linux, and Windows , define a senha que o IBM MQ usa para acessar o repositório de chaves Esse campo é criptografado usando o sistema de proteção de senha

  SSLCRLNL
  O atributo SSLCRLNL especifica uma lista de nomes de objetos de informações sobre autenticação que são usados para fornecer locais de revogação de certificado para permitir verificação aprimorada de certificados TLS.

  SSLCRYP
  Em sistemas AIX, Linux, and Windows, configura o atributo de gerenciador de filas **SSLCryptoHardware** . Esse atributo é o nome da sequência de parâmetros que pode ser usada para configurar o hardware criptográfico que você tem no sistema.

  SSLEV
  Determina se uma mensagem do evento TLS será relatada se um canal que usa TLS falhar ao estabelecer uma conexão TLS.

  SSLFIPS
  Especifica se apenas algoritmos certificados por FIPS devem ser usados se a criptografia for executada no IBM MQ, em vez de no hardware de criptografia. Se o hardware de criptografia for configurado, os módulos de criptografia fornecidos pelo produto de hardware são usados, e estes podem ser certificados por FIPS em um nível específico. Isto depende do produto de hardware em uso.

SSLKEYR
Em sistemas AIX, Linux, and Windows, associa um repositório de chaves a um gerenciador de filas. O GSKit permite usar a segurança TLS em sistemas AIX, Linux, and Windows .

SSLRKEYC
O número de bytes a serem enviados e recebidos dentro de uma conversa TLS antes que a chave secreta seja renegociada. O número de bytes inclui informações de controle enviadas pelo MCA.

- Os parâmetros de canal a seguir suportam TLS:

CERTLABL
Define um rótulo de certificado pessoal a ser usado.

SSLCAUTH
Define se o IBM MQ requer e valida um certificado do cliente TLS.

SSLCIPH
Especifica a segurança da criptografia e a função (CipherSpec), por exemplo, TLS_RSA_WITH_AES_128_CBC_SHA. O CipherSpec deve corresponder em ambas as extremidades do canal.

SSLPEER
Especifica o nome distinto (identificador exclusivo) de parceiros permitidos.

Esta seção descreve os comandos **setmqaut**, **dspmqaut**, **dmpmqaut**, **rcrmqobj**, **rcdmqimg** e **dspmqfls** para suportar o objeto de informações sobre autenticação. Ele também descreve os comandos que podem ser usados para gerenciar chaves e certificados no AIX, Linux, and Windows Consulte as seções a seguir:

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- “Gerenciando chaves e certificados no AIX, Linux, and Windows” na página 549

Para obter uma visão geral de segurança do canal usando TLS, veja

- “Protocolos de segurança TLS no IBM MQ” na página 25

Para obter detalhes de comandos MQSC associados ao TLS, veja

- ALTERAR AUTHINFO
- DEFINIR AUTHINFO
- EXCLUIR AUTHINFO
- EXIBIR AUTHINFO

Para obter detalhes de comandos PCF associados ao TLS, veja

- Mudar, copiar e criar objeto de informações sobre autenticação
- Excluir Objeto de Informações sobre Autenticação
- Investigar Objeto de Informações sobre Autenticação

z/OS *IBM MQ for z/OS server connection channel*

The IBM MQ for z/OS SVRCONN channel is not secure without implementing channel authentication, or adding a security exit using TLS. SVRCONN channels do not have a security exit defined by default.

Security concerns

SVRCONN channels are not secure as initially defined, SYSTEM.DEF.SVRCONN for example. To secure a SVRCONN channel you must set up channel authentication using the SET CHLAUTH command, or install a security exit and implement TLS.

You must use a publicly available sample security exit, write a security exit yourself, or purchase a security exit.

There are several samples available that you can use as a good starting point for writing your own SVRCONN channel security exit.

In IBM MQ for z/OS, the member CSQ4BCX3 in your hlq.SCSQC37S library is a security exit sample written in the C language. Sample CSQ4BCX3 is also shipped pre-compiled in your hlq.SCSQAUTH library.

You can implement the CSQ4BCX3 sample exit by copying the compiled member hlq.SCSQAUTH(CSQ4BCX3) into a load library that is allocated to the CSQXLIB DD in your CHIN Proc. Note that the CHIN requires the load library to be set as "Program Controlled".

Alter your SVRCONN channel to set CSQ4BCX3 as the security exit.

When a client connects using that SVRCONN channel, CSQ4BCX3 will authenticate using the **RemoteUserIdentifier** and **RemotePassword** pair from MQCD or, from IBM MQ for z/OS 9.1.4, the **CSPUserIdPtr** and **CSPPasswordPtr** pair from the MQCSP. If authentication is successful it will copy **RemoteUserIdentifier** into **MCAUserIdentifier**, changing the identity context of the thread.

For Long Term Support and Continuous Delivery before IBM MQ for z/OS 9.1.4, when a client connects using that SVRCONN channel, CSQ4BCX3 will authenticate using the **RemoteUserIdentifier** and **RemotePassword** pair from MQCD. If authentication is successful it will copy **RemoteUserIdentifier** into **MCAUserIdentifier**, changing the identity context of the thread.

If you are writing an IBM MQ Java client you can use pop-ups to query the user and set MQEnvironment.userID and MQEnvironment.password. These values will be passed when the connection is made.

Now that you have a functional security exit, there is the additional concern that the userid and password are being transmitted in plain text across the network when the connection is made, as are the contents of any subsequent IBM MQ messages. You can use TLS to encrypt this initial connection information as well as the contents of any IBM MQ messages.

Example

To secure the IBM MQ Explorer SVRCONN channel SYSTEM.ADMIN.SVRCONN complete the following steps:

1. Copy hlq.SCSQAUTH(CSQ4BCX3) into a load library that is allocated to the CSQXLIB DD in the CHINIT Proc.
2. Verify that load library is Program Controlled.
3. Alter the SYSTEM ADMIN.SVRCONN to use security exit CSQ4BCX3.
4. In IBM MQ Explorer, right-click the z/OS Queue Manager name, select **Connection Details** > **Properties** > **Userid** and enter your z/OS user ID.
5. Connect to the z/OS Queue Manager by entering a password.

Additional information

For exit CSQ4BCX3 to run in a Program Controlled environment, everything loaded into the CHIN address space must be loaded from a Program Controlled library, for example, all libraries in STEPLIB and any

libraries named on CSQXLIB DD. To set a load library as Program Controlled issue RACF commands. In the following example the load library name is MY.TEST.LOADLIB.

```
RALTER PROGRAM * ADDMEM('MY.TEST.LOADLIB'//NOPADCHK)
SETROPTS WHEN(PROGRAM)REFRESH
```

To alter the SVRCONN channel to implement CSQ4BCX3, issue the following IBM MQ command:

```
ALTER CHANNEL(SYSTEM ADMIN.SVRCONN) CHLTYPE(SVRCONN) SCYEXIT(CSQ4BCX3)
```

In the example above, the SVRCONN channel name being used is SYSTEM ADMIN.SVRCONN.

See “Programas de Saída de Canal” on page 115 for more information about channel exits.

Related tasks
Writing channel exit programs on z/OS

serviços de segurança do SNA LU 6.2

O SNA LU 6.2 oferece criptografia em nível de sessão, autenticação em nível de sessão e autenticação em nível de conversa.

Nota: Esta coleção de tópicos supõe que você tenha um entendimento básico de Systems Network Architecture (SNA). A outra documentação referida nesta seção contém uma breve introdução aos conceitos e terminologia relevantes. Se precisar de uma introdução técnica mais abrangente ao SNA, consulte *Systems Network Architecture Technical Overview*, GC30-3073.

O SNA LU 6.2 oferece três serviços de segurança:

- Criptografia em nível de sessão
- Autenticação em nível de sessão
- Autenticação em nível de conversação

Para criptografia em nível de sessão e autenticação em nível de sessão, o SNA utiliza o algoritmo do *DES (Data Encryption Standard)*. O algoritmo do DES é um algoritmo de cifra de bloco, que utiliza uma chave simétrica para criptografar e decriptografar dados. O bloco e a chave têm oito bytes de comprimento.

Criptografia em nível de sessão
A *Criptografia em nível de sessão* criptografa e decriptografa dados de sessão utilizando o algoritmo do DES. Ela pode portanto, ser utilizada para fornecer um serviço de confidencialidade em nível de link em canais do SNA LU 6.2.

LUs (Logical units) podem fornecer criptografia de dados obrigatória (ou necessária), criptografia de dados seletiva ou nenhuma criptografia de dados.

Em uma *sessão criptográfica obrigatória*, uma LU criptografa todas as unidades de pedido de dados de transmissão e decriptografa todas as unidades de pedido de dados de recepção.

Em uma *sessão criptográfica seletiva*, uma LU criptografa apenas as unidades de pedido de dados especificadas pelo TP (transaction program) de envio. A LU de envio indica que os dados são criptografados definindo um indicador no cabeçalho de pedido. Verificando esse indicador, a LU de recebimento pode definir quais unidades de pedido serão decriptografadas antes de transferi-las para o TP de recebimento.

Em uma rede SNA, os MCAs do IBM MQ são programas de transação. Os MCAs não solicitam criptografia para os dados que enviam. Portanto, a criptografia de dados seletiva não é uma opção; apenas a criptografia de dados obrigatória ou nenhuma criptografia de dados é possível em uma sessão.

Para obter informações sobre como implementar a criptografia de dados obrigatória, consulte a documentação para seu subsistema SNA. Consulte a mesma documentação para obter informações sobre formas mais fortes de criptografia que podem estar disponíveis para uso em sua plataforma, como a criptografia Triple DES de 24 bytes no z/OS

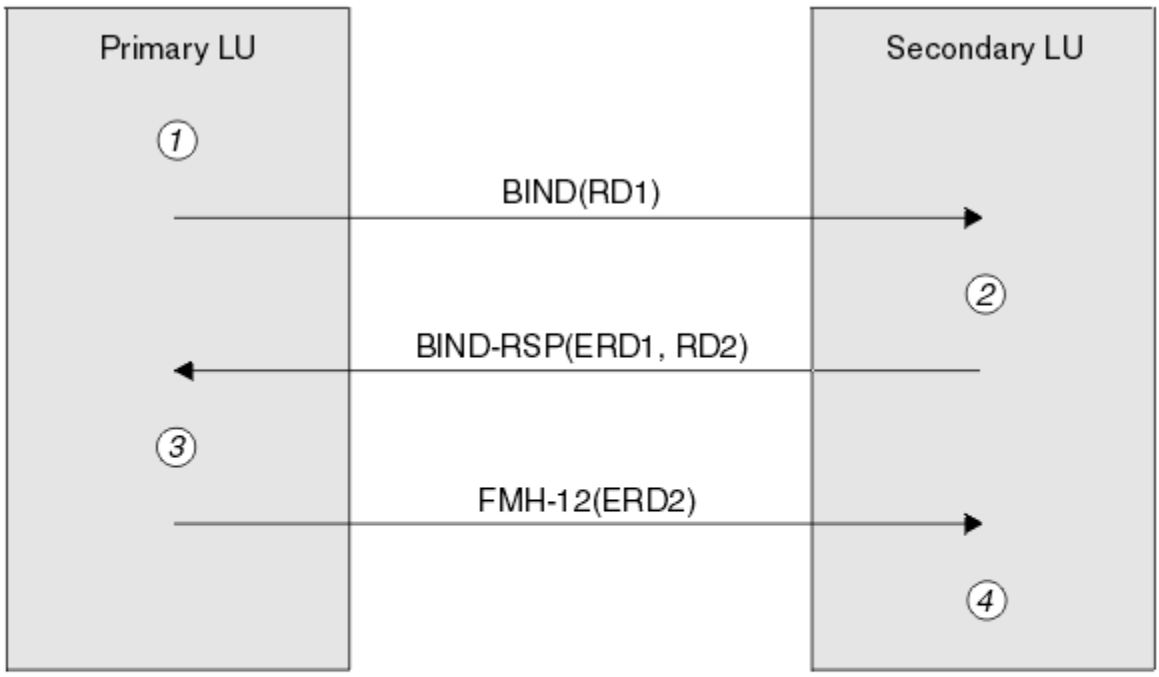
Para obter mais informações gerais sobre criptografia em nível de sessão, consulte *Systems Network Architecture LU 6.2 Reference: Peer Protocols* , SC31-6808.

Autenticação em nível de sessão
A *Autenticação em nível de sessão* é um protocolo de segurança em nível de sessão que permite que duas LUs autentiquem uma à outra enquanto ativam a sessão. Também conhecida como *verificação de LU-LU*.

Como uma LU é efetivamente o "gateway" para um sistema a partir da rede, você pode considerar esse nível de autenticação como suficiente em certas circunstâncias. Por exemplo, se seu gerenciador de filas precisar trocar mensagens com um gerenciador de filas remoto sendo executado em um ambiente controlado e confiável, você pode estar preparado para confiar nas identidades dos componentes restantes do sistema remoto após a autenticação da LU.

A autenticação em nível de sessão é conseguida por cada LU verificando a senha de seu parceiro. A senha é denominada uma *senha de LU-LU* porque é estabelecida uma senha entre cada par de LUs. A maneira que uma senha de LU-LU é estabelecida depende da implementação e está fora do escopo do SNA.

Figura 12 na página 128 ilustra os fluxos para autenticação em nível de sessão.



Legend:

BIND	=	BIND request unit
BIND-RSP	=	BIND response unit
ERD	=	Encrypted random data
FMH-12	=	Function Management Header 12
RD	=	Random data

Figura 12. Fluxos para autenticação em nível de sessão

O protocolo para autenticação em nível de sessão é o seguinte. Os números no procedimento correspondem aos números em Figura 12 na página 128.

1. A LU primária gera um valor de dados aleatório (RD1) e o envia para a LU secundária no pedido BIND.
2. Quando a LU secundária recebe o pedido BIND com os dados aleatórios, ela criptografa os dados utilizando o algoritmo do DES com sua cópia da senha de LU-LU como a chave. A LU secundária, então, gera um segundo valor de dados aleatórios (RD2) e envia-o com os dados criptografados (ERD1) para a LU primária na resposta BIND.

3. Quando a LU primária recebe a resposta BIND, ela calcula sua própria versão dos dados criptografados a partir dos dados aleatórios gerados originalmente. Ela o faz utilizando o algoritmo do DES com sua cópia da senha de LU-LU como a chave. Então ela compara sua versão com os dados criptografados recebidos na resposta BIND. Se os dois valores forem iguais, a LU primária saberá que a LU secundária tem a mesma senha que ela e a LU secundária será autenticada. Se os dois valores não corresponderem, a LU primária encerrará a sessão.

   A LU primária criptografa os dados aleatórios recebidos na resposta BIND e envia os dados criptografados (ERD2) para a LU secundária em um FMH-12 (Function Management Header 12).
4. Quando a LU secundária recebe o FMH-12, ela calcula sua própria versão dos dados criptografados a partir dos dados aleatórios gerados por ela. Então ela compara sua versão com os dados criptografados recebidos no FMH-12. Se os dois valores forem iguais, a LU primária será autenticada. Se os dois valores não corresponderem, a LU secundária encerrará a sessão.

Em uma versão aperfeiçoada do protocolo, que fornece melhor proteção contra ataques humano intermediários, a LU secundária calcula um DES MAC (Message Authentication Code) a partir de RD1, RD2 e o nome completo da LU secundária, utilizando sua cópia da senha de LU-LU como a chave. A LU secundária envia o MAC para a LU primária na resposta BIND em vez de ERD1.

A LU primária autentica a LU secundária calculando sua própria versão do MAC, a qual ela compara com o MAC recebido na resposta BIND. Em seguida a LU primária calcula um segundo MAC a partir de RD1 e RD2, e envia o MAC para a LU secundária no FMH-12 em vez do ERD2.

A LU secundária autentica a LU primária calculando sua própria versão do segundo MAC, a qual ela compara com o MAC recebido no FMH-12.

Para obter informações sobre como configurar a autenticação em nível de sessão, consulte a documentação para seu subsistema SNA. Para obter mais informações gerais sobre autenticação de nível de sessão, consulte *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

Autenticação em nível de conversação

Quando um TP local tenta alocar uma conversação com um TP parceiro, a LU local envia um pedido de anexo para a LU parceira, pedindo que anexe o TP parceiro. Em certas circunstâncias, o pedido de anexo pode conter informações de segurança que a LU parceira pode utilizar para autenticar o TP local. Isso é conhecido como *autenticação em nível de conversação* ou *verificação do usuário final*.

Os tópicos a seguir descrevem como o IBM MQ fornece suporte para autenticação em nível de conversação.

Para obter mais informações sobre autenticação em nível de conversação, consulte *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

z/OS Para obter informações específicas para z/OS, consulte z/OS Planejamento de MVS: Gerenciamento de APPC/MVS.

Para obter mais informações sobre CPI-C, consulte Usando Comunicações de CPI.

Para obter mais informações sobre APPC/MVS TP Conversation Callable Services, consulte APPC/MVS TP Conversation Callable Services.

Multi *Suporte para autenticação de nível de conversação em multiplataformas*

Use este tópico para obter uma visão geral de como funciona a autenticação de nível de conversação em multiplataformas.

O suporte para autenticação de nível de conversação em multiplataformas é ilustrado na Figura 13 na página 130. Os números no diagrama correspondem aos números na descrição a seguir.

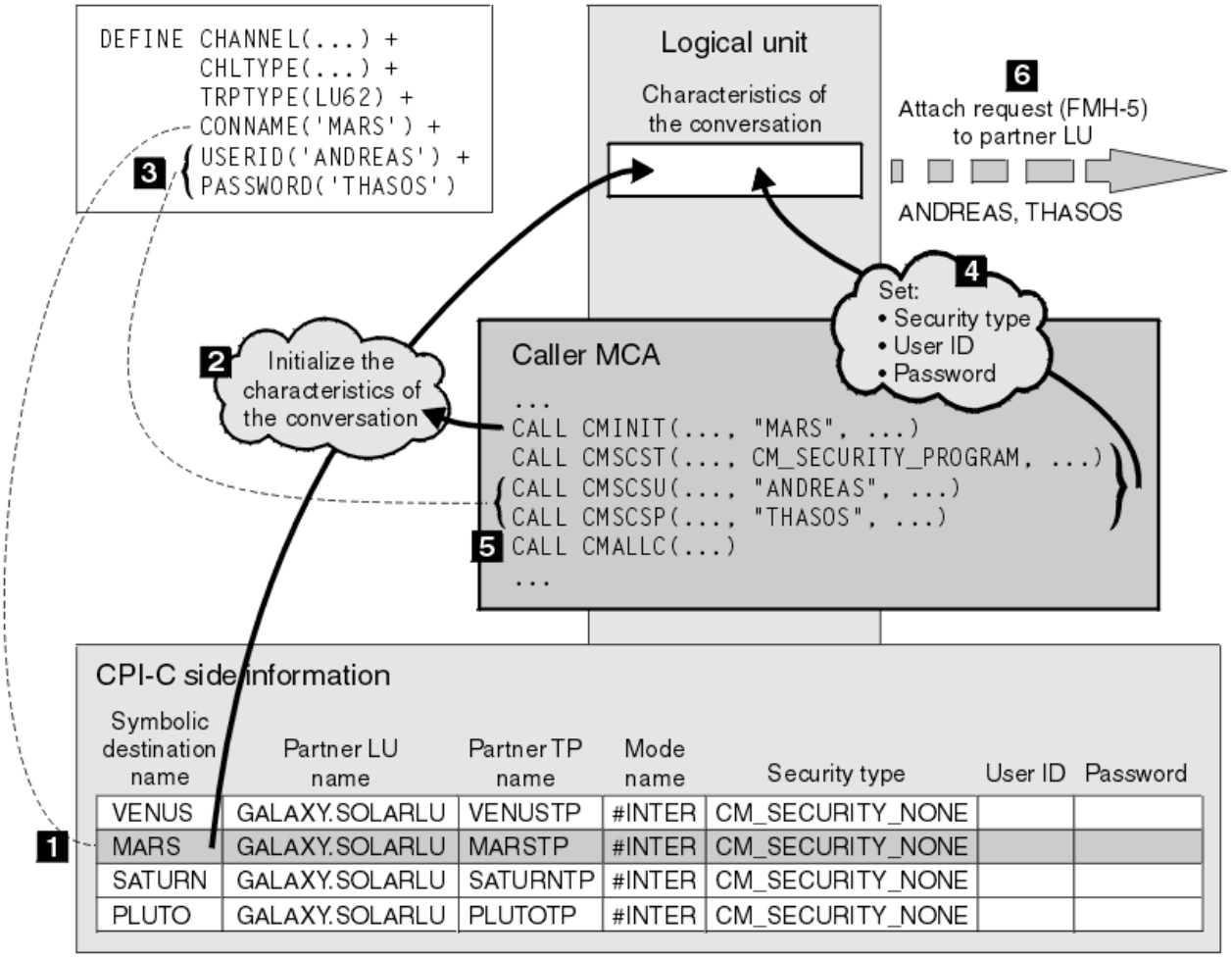


Figura 13. O suporte do IBM MQ para autenticação em nível de conversa

Em multiplataformas, um MCA usa chamadas de Comunicações da Interface de Programação Comum (CPI-C) para se comunicar com um MCA parceiro por meio de uma rede SNA. Na definição de canal na extremidade do responsável pela chamada de um canal, o valor do parâmetro CONNAME é um nome de destino simbólico, que identifica uma entrada de informações secundárias de CPI-C (1). Essa entrada especifica:

- O nome da LU parceira
- O nome do TP parceiro, que é um MCA receptor da chamada
- O nome do modo a ser utilizado para a conversação

Uma entrada de informações secundárias também pode especificar as seguintes informações de segurança:

- Um tipo de segurança.

  Os tipos de segurança geralmente implementados são CM_SECURITY_NONE, CM_SECURITY_PROGRAM e CM_SECURITY_SAME, mas outros são definidos na especificação de CPI-C.
- Um ID do usuário.
- Uma senha.

Um MCA originador da chamada se prepara para alocar uma conversação com um MCA receptor da chamada emitindo a chamada de CPI-C CMINIT, utilizando o valor de CONNAME como um dos parâmetros na chamada. A chamada CMINIT identifica, para o benefício da LU local, a entrada de informações secundárias que o MCA pretende utilizar para a conversação. A LU local usa os valores nesta entrada para inicializar as características da conversa (2).

O MCA do responsável pela chamada verifica os valores dos parâmetros USERID e PASSWORD na definição de canal (3). Se USERID for definido, o MCA do responsável pela chamada emite as seguintes chamadas de CPI-C (4):

- CMSCST, para definir o tipo de segurança para a conversação para CM_SECURITY_PROGRAM.
- CMSCSU, para definir o ID de usuário para a conversação para o valor de USERID.
- CMSCSP, para definir a senha para a conversação para o valor de PASSWORD. CMSCSP não é chamado a menos que PASSWORD seja definido.

O tipo de segurança, ID de usuário e senha definidos por essas chamadas substituem quaisquer valores adquiridos anteriormente da entrada de informações secundárias.

O MCA do responsável pela chamada emite a chamada de CPI-C CMALLC para alocar a conversa (5). Em resposta a essa chamada, a LU local envia uma solicitação de conexão (Function Management Header 5 ou FMH-5) para a LU do parceiro (6).

Se a LU parceira aceitar um ID de usuário e uma senha, os valores de USERID e de PASSWORD serão incluídos no pedido de anexo. Se a LU parceira não aceitar um ID de usuário e uma senha, os valores não serão incluídos no pedido de anexo. A LU local descobre se a LU parceira aceitará um ID de usuário e uma senha como parte de uma troca de informações quando as LUs se ligarem para formar uma sessão.

Em uma versão posterior do pedido de anexo, um substituto de senha pode fluir entre as LUs em vez de uma senha propriamente dita. Um substituto de senha é um MAC (Message Authentication Code) ou uma compilação de mensagens SHA-1, formados a partir da senha. Substitutos de senha podem ser utilizados apenas se ambas as LUs os suportarem.

Quando a LU parceira recebe um pedido de anexo de entrada contendo um ID de usuário e uma senha, ela pode utilizar o ID de usuário e a senha para os objetivos de identificação e de autenticação. Referindo-se às listas de controle de acesso, a LU parceira também pode determinar se o ID de usuário tem autoridade para alocar uma conversação e anexar o MCA receptor da mensagem.

Além disso, o MCA receptor da mensagem pode ser executado sob o ID de usuário incluído no pedido de anexo. Nesse caso, o ID de usuário se torna o ID de usuário padrão para o MCA receptor da chamada e é utilizado para verificações de autoridade quando o MCA tenta se conectar ao gerenciador de filas. Ele também pode ser utilizado para verificações de autoridade subseqüentemente quando o MCA tenta acessar os recursos do gerenciador de filas.

A forma como um ID de usuário e uma senha em um pedido de anexo podem ser utilizados para identificação, autenticação e controle de acesso depende da implementação. Para obter informações específicas para seu subsistema SNA, consulte a documentação apropriada.

Se USERID não for definido, o MCA originador da chamada não chamará CMSCST, CMSCSU e CMSCSP. Nesse caso, as informações de segurança que circulam em um pedido de anexo são determinadas unicamente pelo que for especificado na entrada de informações secundárias e o que a LU parceira irá aceitar.

Conversation level authentication and IBM MQ for z/OS
Use this topic to gain an overview of how conversation level authentication works, on z/OS.

On IBM MQ for z/OS, MCAs do not use CPI-C. Instead, they use APPC/MVS TP Conversation Callable Services, an implementation of Advanced Program-to-Program Communication (APPC), which has some CPI-C features. When a caller MCA allocates a conversation, a security type of SAME is specified on the call. Therefore, because an APPC/MVS LU supports persistent verification only for inbound conversations, not for outbound conversations, there are two possibilities:

- If the partner LU trusts the APPC/MVS LU and will accept an already verified user ID, the APPC/MVS LU sends an attach request containing:
  - The channel initiator address space user ID
  - A security profile name, which, if RACF is used, is the name of the current connect group of the channel initiator address space user ID
  - An already verified indicator

- If the partner LU does not trust the APPC/MVS LU and will not accept an already verified user ID, the APPC/MVS LU sends an attach request containing no security information.

On IBM MQ for z/OS, the USERID and PASSWORD parameters on the DEFINE CHANNEL command cannot be used for a message channel and are valid only at the client connection end of an MQI channel. Therefore, an attach request from an APPC/MVS LU never contains values specified by these parameters.

Segurança para Clusters de Gerenciadores de Filas

Embora o uso dos clusters de gerenciadores de filas possa ser conveniente, você deve prestar muita atenção à sua segurança.

Um *cluster de gerenciadores de filas* é uma rede de gerenciadores de filas que estão associados logicamente de alguma maneira. Um gerenciador de filas que seja um membro de um cluster é chamado um *gerenciador de filas de cluster*.

Uma fila que pertença a um gerenciador de filas de cluster pode ser tornada conhecida a outros gerenciadores de filas no cluster. Uma fila assim é chamada uma *fila de cluster*. Qualquer gerenciador de filas em um cluster pode enviar mensagens para filas do cluster sem precisar de nenhum dos seguintes:

- Uma definição explícita de fila remota para cada fila do cluster
- Canais explicitamente definidos para e de cada gerenciador de filas remotas
- Uma fila de transmissão separada para cada canal de transmissão

É possível criar um cluster no qual dois ou mais gerenciadores de filas são clones. Isso significa que eles possuem ocorrências das mesmas filas locais, incluindo quaisquer filas locais declaradas como filas de cluster, e podem suportar ocorrências dos mesmos aplicativos do servidor.

Quando um aplicativo conectado a um gerenciador de filas do cluster envia uma mensagem a uma fila de clusters que tem uma instância em cada um dos gerenciadores de filas clonados, o IBM MQ decide para qual gerenciador de filas enviá-la. Quando muitos aplicativos enviam mensagens para a fila de clusters, o IBM MQ equilibra a carga de trabalho entre todos os gerenciadores de filas que possuem uma instância da fila. Se um dos sistemas que hospedam um gerenciador de filas clonado falhar, o IBM MQ continuará a equilibrar a carga de trabalho entre os gerenciadores de filas restantes até que o sistema que falhou seja reiniciado.

Se você estiver utilizando clusters de gerenciadores de filas, precisará levar em consideração as seguintes questões de segurança:

- Permitir que somente gerenciadores de filas selecionados enviem mensagens a seu gerenciador de filas
- Permitir que somente usuários selecionados de um gerenciador de filas remoto enviem mensagens a uma fila no seu gerenciador de filas
- Permitir que aplicativos conectados a seu gerenciador de filas enviem mensagens somente a filas remotas selecionadas

Essas considerações são relevantes mesmo que você não esteja utilizando clusters, mas se tornam mais importantes se estiverem sendo utilizados clusters.

Se um aplicativo puder enviar mensagens a uma fila de cluster, ele poderá enviar mensagens a qualquer outra fila do cluster sem precisar de definições adicionais de filas remotas, filas de transmissão ou canais. Portanto, torna-se mais importante considerar se é preciso restringir o acesso às filas do cluster em seu gerenciador de filas, e restringir as filas do cluster às quais os aplicativos podem enviar mensagens.

Existem algumas considerações de segurança adicionais que são relevantes somente se você estiver utilizando clusters de gerenciadores de filas:

- Permitir que somente gerenciadores de filas selecionados se unam a um cluster
- Forçando Gerenciadores de Filas Indesejáveis a Sair de um Cluster

Para obter mais informações sobre todas essas considerações, consulte Mantendo os Clusters Seguros.

z/OS Para considerações específicas para o IBM MQ for z/OS, consulte “Security in queue manager clusters on z/OS” na página 267.

Tarefas relacionadas

"Impedindo que gerenciadores de filas recebam mensagens" na página 489
É possível evitar que um gerenciador de filas do cluster receba mensagens que ele não está autorizado a receber, usando programas de saída.

Segurança para o Publicar/assinar do IBM MQ

Há considerações de segurança adicionais se você estiver usando o publicar/assinar do IBM MQ.

Em um sistema de publicação/assinatura, há dois tipos de aplicativo: publicador e assinante. Os *publicadores* fornecem informações no formato de mensagens do IBM MQ. Quando um publicador publica uma mensagem, ele especifica um *tópico*, que identifica o assunto das informações dentro da mensagem.

Assinantes são os consumidores das informações que são publicadas. Um assinante especifica os tópicos nos quais está interessado inscrevendo-se neles.

O *gerenciador de filas* é um aplicativo fornecido com o Publicar/assinar do IBM MQ. Ele recebe as mensagens publicadas dos publicadores e pedidos de assinatura dos assinantes, e encaminha as mensagens publicadas aos assinantes. São enviadas a um assinante somente as mensagens sobre os tópicos que ele assinou.

Para obter mais informações, consulte Segurança da Publicação/Assinatura.

Segurança de multicast

Use estas informações para entender por que os processos de segurança podem ser necessários com o IBM MQ Multicast.

O IBM MQ Multicast não tem segurança integrada. As verificações de segurança são manipuladas no gerenciador de filas no tempo de MQOPEN, e a configuração do campo MQMD é manipulada pelo cliente. Alguns aplicativos na rede podem não ser os aplicativos IBM MQ (Por exemplo, os aplicativos LLM, consulte Interoperabilidade multicast com IBM MQ Sistema de mensagens de baixa latência para obter mais informações), portanto, você pode precisar implementar seus próprios procedimentos de segurança porque os aplicativos de recebimento não podem ter certeza da validade dos campos de contexto.

Há três processos de segurança a serem considerados:

Controle de acesso

O controle de acesso em IBM MQ é baseado nos IDs do usuário. Para obter informações adicionais sobre este assunto, consulte "Controle de acesso para clientes" na página 107.

Segurança de rede

Uma rede isolada pode ser uma opção de segurança viável para evitar mensagens falsas. É possível para um aplicativo no endereço de grupo multicast publicar mensagens maliciosas usando as funções de comunicação nativa, que são indistinguíveis a partir das mensagens do MQ por virem de um aplicativo no mesmo endereço de grupo multicast.

Também é possível para um cliente no endereço de grupo multicast receber mensagens que foram destinadas a outros clientes no mesmo endereço de grupo multicast.

Isolar a rede multicast assegura que apenas clientes e aplicativos válidos possuam acesso. Esta precaução de segurança pode impedir as mensagens maliciosas de entrar e as informação confidenciais de sair.

Para obter informações sobre endereços de rede de grupo multicast, consulte: Configurando a rede apropriada para tráfego multicast

Assinaturas Digitais

Uma assinatura digital é formada pela criptografia de uma representação de uma mensagem. A criptografia utiliza a chave privada do assinante e, para eficiência, geralmente opera em uma compilação de mensagens, ao invés de na mensagem em si. Assinar digitalmente uma mensagem

antes de um MQPUT é uma boa precaução de segurança, mas esse processo pode ter um efeito negativo no desempenho se houver um grande volume de mensagens.

As assinaturas digitais variam com os dados que estão sendo assinados. Se duas mensagens diferentes forem assinadas digitalmente pela mesma entidade, as duas assinaturas diferirão, mas ambas poderão ser verificadas com a mesma chave pública, ou seja, a chave pública da entidade que assinou as mensagens.

Conforme mencionado anteriormente nesta seção, pode ser possível para um aplicativo no endereço de grupo multicast publicar mensagens maliciosas usando funções de comunicação nativa, que são indistinguíveis a partir de mensagens do MQ. As assinaturas digitais fornecem prova de origem, e somente o emissor conhece a chave privada, que fornece forte evidência de que o emissor é o originador da mensagem.

Para obter informações adicionais sobre este assunto, consulte "Conceitos criptográficos" na página 11.

Firewalls e IBM MQ Internet Pass-Thru

O IBM MQ Internet Pass-Thru pode simplificar a comunicação por meio de um firewall

O MQIPT permite que dois gerenciadores de filas troquem mensagens ou um aplicativo cliente IBM MQ se conecte a um gerenciador de filas, sem requerer uma conexão TCP/IP direta. Essa arquitetura é útil se um firewall proibir uma conexão TCP/IP direta entre dois sistemas... Usar MQIPT como um proxy pode tornar a passagem de dados do canal IBM MQ por meio de um firewall mais simples e gerenciável. O MQIPT também pode proteger os dados do IBM MQ que são enviados pela Internet usando a Segurança da camada de transporte (TLS) e os dados do túnel IBM MQ dentro do HTTP.

Para obter mais informações, consulte IBM MQ Internet Pass-Thru.

z/OS IBM MQ for z/OS security implementation checklist

This topic gives a step-by-step procedure you can use to work out and define the security implementation for each of your IBM MQ queue managers.

RACF provides definitions for the IBM MQ security classes in its supplied static Class Descriptor Table (CDT). As you work through the checklist, you can determine which of these classes your setup requires. You must ensure that they are activated as described in "RACF security classes" on page 193.

Refer to other sections for details, in particular "Profiles used to control access to IBM MQ resources" on page 203.

If you require security checking, follow this checklist to implement it:

1. Activate the RACF MQADMIN (uppercase profiles) or MXADMIN (mixed case profiles) class.
   - Do you want security at queue sharing group level, queue manager level, or a combination of both?

     See, "Profiles to control queue sharing group or queue manager level security" on page 198.
2. Do you need connection security?
   - **Yes**: Activate the MQCONN class. Define appropriate connection profiles at either queue manager level or queue sharing group level in the MQCONN class. Then permit the appropriate users or groups access to these profiles.

     Note: Only users of the `MQCONN` API request or CICS or IMS address space user IDs need to have access to the corresponding connection profile.
   - **No**: Define an hlq.NO.CONNECT.CHECKS profile at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class.
3. Do you need security checking on commands?
   - **Yes**: Activate the MQCMDS class. Define appropriate command profiles at either queue manager level or queue sharing group level in the MQCMDS class. Then permit the appropriate users or groups access to these profiles.

If you are using a queue sharing group, you might need to include the user IDs used by the queue manager itself and the channel initiator. See "Setting up IBM MQ for z/OS resource security" on page 259.

- **No**: Define an hlq.NO.CMD.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.

4. Do you need security on the resources used in commands?
   - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define appropriate profiles for protecting resources on commands at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the appropriate users or groups access to these profiles. Set the CMDUSER parameter in CSQ6SYSP to the default user ID to be used for command security checks.

     If you are using a queue sharing group, you might need to include the user IDs used by the queue manager itself and the channel initiator. See "Setting up IBM MQ for z/OS resource security" on page 259.
   - **No**: Define an hlq.NO.CMD.RESC.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
5. Do you need queue security?
   - **Yes**: Activate the MQQUEUE or MXQUEUE class. Define appropriate queue profiles for the required queue manager or queue sharing group in the MQQUEUE or MXQUEUEclass. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.QUEUE.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
6. Do you need process security?
   - **Yes**: Activate the MQPROC or MXPROC class. Define appropriate process profiles at either queue manager or queue sharing group level and permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.PROCESS.CHECKS profile for the appropriate queue manager or queue sharing group in the MQADMIN or MXADMIN class.
7. Do you need namelist security?
   - **Yes**: Activate the MQNLIST or MXNLISTclass. Define appropriate namelist profiles at either queue manager level or queue sharing group level in the MQNLIST or MXNLIST class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.NLIST.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
8. Do you need topic security?
   - **Yes**: Activate the MXTOPIC class. Define appropriate topic profiles at either queue manager level or queue sharing group level in the MXTOPIC class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.TOPIC.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
9. Do any users need to protect the use of the `MQOPEN` or `MQPUT1` options relating to the use of context?
   - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define hlq.CONTEXT.queuename profiles at the queue, queue manager, or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the appropriate users or groups access to these profiles.
   - **No**: Define an hlq.NO.CONTEXT.CHECKS profile for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.
10. Do you need to protect the use of alternative user IDs?

- **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define the appropriate hlq.ALTERNATE.USER. *alternateuserid* profiles for the required queue manager or queue sharing group and permit the required users or groups access to these profiles.
- **No**: Define the profile hlq.NO.ALTERNATE.USER.CHECKS for the required queue manager or queue sharing group in the MQADMIN or MXADMIN class.

11. Do you need to tailor which user IDs are to be used for resource security checks through RESLEVEL?
    - **Yes**: Ensure the MQADMIN or MXADMIN class is active. Define an hlq.RESLEVEL profile at either queue manager level or queue sharing group level in the MQADMIN or MXADMIN class. Then permit the required users or groups access to the profile.
    - **No**: Ensure that no generic profiles exist in the MQADMIN or MXADMIN class that can apply to hlq.RESLEVEL. Define an hlq.RESLEVEL profile for the required queue manager or queue sharing group and ensure that no users or groups have access to it.
12. Do you need to 'timeout' unused user IDs from IBM MQ ?
    - **Yes**: Determine what timeout values you would like to use and issue the MQSC ALTER SECURITY command to change the TIMEOUT and INTERVAL parameters.
    - **No**: Issue the MQSC ALTER SECURITY command to set the INTERVAL value to zero.

    Note: Update the CSQINP1 initialization input data set used by your subsystem so that the MQSC ALTER SECURITY command is issued automatically when the queue manager is started.
13. Do you use distributed queuing?
    - **Yes**: Use channel authentication records. For more information, see “Registros de Autenticação de Canal” on page 53.
    - You can also determine the appropriate MCAUSER attribute value for each channel, or provide suitable channel security exits.
14. Do you want to use Transport Layer Security (TLS)?
    - **Yes**: To specify that any user presenting an TLS personal certificate containing a specified DN is to use a specific MCAUSER, set a channel authentication record of type SSLPEERMAP. You can specify a single distinguished name or a pattern including wildcards.
    - Plan your TLS infrastructure. Install the System SSL feature of z/OS. In RACF, set up your certificate name filters (CNFs), if you are using them, and your digital certificates. Set up your SSL key ring. Ensure that the SSLKEYR queue manager attribute is nonblank and points to your SSL key ring. Also ensure that the value of SSLTASKS is at least 2.
    - **No**: Ensure that SSLKEYR is blank, and SSLTASKS is zero.

    For further details about TLS, see “Protocolos de segurança TLS no IBM MQ” on page 25.
15. Do you use clients?
    - **Yes**: Use channel authentication records.
    - You can also determine the appropriate MCAUSER attribute value for each server-connection channel, or provide suitable channel security exits if required.
16. Check your switch settings.

    IBM MQ issues messages when the queue manager is started that display your security settings. Use these messages to determine whether your switches are set correctly.
17. Do you send passwords from client applications?
    - **Yes**: Ensure that the z/OS feature is installed and Integrated Cryptographic Service Facility (ICSF) is started for the best protection.
    - **No**: You can ignore the error message reporting that ICSF has not started.

    For further information about ICSF see “Using the Integrated Cryptographic Service Facility (ICSF)” on page 267

Configurar a segurança

Esta coleção de tópicos contém informações específicas para sistemas operacionais diferentes e para a utilização de clientes.

ALW Configurando a Segurança em AIX, Linux, and Windows

Considerações de segurança específicas para sistemas AIX, Linux, and Windows .

Gerenciadores de filas do IBM MQ transferem informações que são potencialmente de valor, portanto, é necessário usar um sistema de autoridade para assegurar que usuários não autorizados não possam acessar seus gerenciadores de filas. Considere os seguintes tipos de controles de segurança:

Quem pode administrar o IBM MQ
É possível definir o conjunto de usuários que pode emitir comandos para administrar o IBM MQ.

Quem pode usar objetos do IBM MQ
É possível definir quais usuários (geralmente aplicativos) podem usar chamadas MQI e comando de PCF para fazer o seguinte:

- Quem pode se conectar a um gerenciador de filas.
- Quem pode acessar objetos (filas, definições de processos, listas de nomes, canais, canais de conexão do cliente, listeners, serviços e objetos de informações sobre autenticação) e que tipo de acesso eles têm a esses objetos.
- Quem pode acessar mensagens do IBM MQ.
- Quem pode acessar as informações de contexto associadas a uma mensagem.

Segurança de canal
É necessário assegurar que os canais usados para enviar mensagens para sistemas remotos possam acessar os recursos necessários.

É possível usar recursos de operação padrão para conceder acesso a bibliotecas de programas, bibliotecas de link do MQI e comandos. No entanto, o diretório que contém as filas e outros dados do gerenciador de filas é privado ao IBM MQ; não use comandos do sistema operacional padrão para conceder ou revogar autorizações para recursos do MQI.

ALW Como as Autorizações funcionam no AIX, Linux, and Windows

As tabelas de especificação de autorização nos tópicos desta seção definem precisamente como as autorizações funcionam e as restrições que se aplicam.

As tabelas aplicam-se a estas situações:

- Aplicativos que emitem chamadas MQI
- Programas de administração que emitem comandos MQSC como PCFs Escape
- Programas de administração que emitem comando de PCF

Nesta seção, as informações são apresentadas como um conjunto de tabelas que especificam o seguinte:

Ação a ser executada
Opção de MQI, comando MQSC ou comando PCF.

Objeto de controle de acesso
Fila, processo, gerenciador de filas, lista de nomes, informações sobre autenticação, canal, canal de conexão do cliente, listener ou serviço.

Authorization required
Expressa como uma constante MQZAO_.

Nas tabelas, as constantes prefixadas com MQZAO_ correspondem às palavras-chave na lista de autorização para o comando `setmqaut` da entidade específica. Por exemplo, MQZAO_BROWSE corresponde à palavra-chave `+browse`, MQZAO_SET_ALL_CONTEXT corresponde à palavra-chave

+setall, etc. Essas constantes são definidas no arquivo de cabeçalho cmqzc.h, fornecido com o produto.

ALW *Autorizações para Chamadas MQI*

MQCONN, **MQOPEN**, **MQPUT1** e **MQCLOSE** podem requerer verificações de autorização. As tabelas deste tópico resumem as autorizações necessárias para cada chamada.

Um aplicativo terá permissão para emitir chamadas MQI e opções específicas somente se o identificador de usuário sob o qual estiver sendo executado (ou cujas autorizações puder assumir) tiver recebido a autorização relevante.

Quatro chamadas MQI podem requerer verificações de autorização: **MQCONN**, **MQOPEN**, **MQPUT1** e **MQCLOSE**.

Para **MQOPEN** e **MQPUT1**, a verificação de autoridade é feita no nome do objeto que está sendo aberto e não no nome ou nomes resultantes após a resolução de um nome. Por exemplo, um aplicativo pode receber autoridade para abrir uma fila de alias sem ter autoridade para abrir a fila de base para a qual o alias é resolvido. A regra é que a verificação seja realizada na primeira definição encontrada durante o processo de resolução de um nome que não é um alias do gerenciador de filas, a menos que a definição de alias do gerenciador de filas seja aberta diretamente; ou seja, seu nome é exibido no campo *ObjectName* do descritor de objeto. A autoridade é sempre necessária para o objeto que está sendo aberto. Em alguns casos, autoridade adicional independente da fila, obtida por meio de uma autorização para o objeto de gerenciador de filas, é necessária.

Tabela 10 na página 138, Tabela 11 na página 138, Tabela 12 na página 139 e Tabela 13 na página 140 resumem as autorizações necessárias para cada chamada. Nas tabelas, *Não aplicável* significa que a verificação de autorização não é relevante para essa operação; *Nenhuma verificação* significa que nenhuma verificação de autorização é executada.

Nota: Você não encontrará nenhuma menção a listas de nomes, canais, canais de conexão do cliente, listeners, serviços ou objetos de informações sobre autenticação nessas tabelas. Isso é porque nenhuma das autorizações se aplica a esses objetos, exceto MQOO_INQUIRE, para o qual as mesmas autorizações se aplicam como para os outros objetos.

A autorização especial MQZAO_ALL_MQI inclui todas as autorizações nas tabelas que são relevantes ao tipo de objeto, exceto MQZAO_DELETE e MQZAO_DISPLAY, que são classificados como autorizações de administração.

Para modificar qualquer uma das opções de contexto da mensagem, deve-se ter as autorizações apropriadas para emitir a chamada. Por exemplo, para usar MQOO_SET_IDENTITY_CONTEXT ou MQPMO_SET_IDENTITY_CONTEXT, deve-se ter a permissão +setid.

Tabela 10. Autorização de segurança necessária para chamadas MQCONN

Autorização necessária para:	**Objeto da fila (“1” na página 140)**	**Objeto de processo**	**Objeto do gerenciador de filas**
MQCONN	Não-aplicável	Não-aplicável	MQZAO_CONNECT

Tabela 11. Autorização de segurança necessária para chamadas MQOPEN

Autorização necessária para:	**Objeto da fila (“1” na página 140)**	**Objeto de processo**	**Objeto do gerenciador de filas**
MQOO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE
MQOO_BROWSE	MQZAO_BROWSE	Não-aplicável	Sem verificação
MQOO_INPUT_*	MQZAO_INPUT	Não-aplicável	Sem verificação
MQOO_SAVE_ALL_CONTEXT **(“2” na página 140)**	MQZAO_INPUT	Não-aplicável	Não-aplicável

Tabela 11. Autorização de segurança necessária para chamadas MQOPEN (continuação)

Autorização necessária para:	Objeto da fila (**"1" na página 140**)	Objeto de processo	Objeto do gerenciador de filas
MQOO_OUTPUT (fila Normal) (**"3" na página 140**)	MQZAO_OUTPUT	Não-aplicável	Não-aplicável
MQOO_PASS_ IDENTITY_CONTEXT (**"4" na página 140**)	MQZAO_PASS_ IDENTITY_CONTEXT	Não-aplicável	Sem verificação
MQOO_PASS_ALL_ CONTEXT (**"4" na página 140, "5" na página 140**)	MQZAO_PASS _ALL_CONTEXT	Não-aplicável	Sem verificação
MQOO_SET_ IDENTITY_CONTEXT (**"4" na página 140, "5" na página 140**)	MQZAO_SET_ IDENTITY_CONTEXT	Não-aplicável	MQZAO_SET_ IDENTITY_CONTEXT (**"6" na página 140**)
MQOO_SET_ ALL_CONTEXT (**"4" na página 140, "7" na página 140**)	MQZAO_SET_ ALL_CONTEXT	Não-aplicável	MQZAO_SET_ ALL_CONTEXT (**"6" na página 140**)
MQOO_OUTPUT (Fila de transmissão) (**"8" na página 140**)	MQZAO_SET_ ALL_CONTEXT	Não-aplicável	MQZAO_SET_ ALL_CONTEXT (**"6" na página 140**)
MQOO_SET	MQZAO_SET	Não-aplicável	Sem verificação
MQOO_ALTERNATE_ USER_AUTHORITY	(**"9" na página 140**)	(**"9" na página 140**)	MQZAO_ALTERNATE_ USER_AUTHORITY (**"9" na página 140, "10" na página 140**)

Tabela 12. Autorização de segurança necessária para chamadas MQPUT1

Autorização necessária para:	Objeto da fila (**"1" na página 140**)	Objeto de processo	Objeto do gerenciador de filas
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT (**"11" na página 140**)	Não-aplicável	Sem verificação
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT (**"11" na página 140**)	Não-aplicável	Sem verificação
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT (**"11" na página 140**)	Não-aplicável	MQZAO_SET_ IDENTITY_CONTEXT (**"6" na página 140**)
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT (**"11" na página 140**)	Não-aplicável	MQZAO_SET_ ALL_CONTEXT (**"6" na página 140**)
(Fila de transmissão) (**"8" na página 140**)	MQZAO_SET_ ALL_CONTEXT	Não-aplicável	MQZAO_SET_ ALL_CONTEXT (**"6" na página 140**)

Tabela 12. Autorização de segurança necessária para chamadas MQPUT1 (continuação)			
Autorização necessária para:	**Objeto da fila (“1” na página 140)**	**Objeto de processo**	**Objeto do gerenciador de filas**
MQPMO_ALTERNATE_USER_AUTHORITY	**(“12” na página 140)**	Não-aplicável	MQZAO_ALTERNATE_USER_AUTHORITY **(“10” na página 140)**

Tabela 13. Autorização de segurança necessária para chamadas MQCLOSE			
Autorização necessária para:	**Objeto da fila (“1” na página 140)**	**Objeto de processo**	**Objeto do gerenciador de filas**
MQCO_DELETE	MQZAO_DELETE **(“13” na página 140)**	Não-aplicável	Não-aplicável
MQCO_DELETE _PURGE	MQZAO_DELETE **(“13” na página 140)**	Não-aplicável	Não-aplicável

Notas para as tabelas:

1. Se for abrir uma fila modelo:
   - A autoridade MQZAO_DISPLAY será necessária para a fila modelo, além da autoridade para abrir a fila modelo para o tipo de acesso para o qual você está abrindo.
   - A autoridade MQZAO_CREATE não é necessária para criar o fila dinâmica.
   - O identificador de usuários usado para abrir a fila modelo recebe automaticamente todas as autoridades específicas da fila (equivalente a MQZAO_ALL) para a fila dinâmica criada.
2. MQOO_INPUT_* também deve ser especificado. Isso é válido para uma fila local, modelo ou de alias.
3. Essa verificação é executada para todos os casos de saída, exceto para filas de transmissão (consulte a nota “8” na página 140).
4. MQOO_OUTPUT também deve ser especificado.
5. MQOO_PASS_IDENTITY_CONTEXT também é sugerido por essa opção.
6. Essa autoridade é necessária para o objeto de gerenciador de filas e a fila específica.
7. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT e MQOO_SET_IDENTITY_CONTEXT também são sugeridos por essa opção.
8. Essa verificação é executada para uma fila local ou modelo que tem um atributo de fila *Usage* de MQUS_TRANSMISSION e está sendo aberta diretamente para saída. Ela não será aplicada se uma fila remota estiver sendo aberta (especificando-se os nomes do gerenciador de filas remotas e da fila remota ou especificando-se o nome de uma definição local da fila remota).
9. Pelo menos um de MQOO_INQUIRE (para qualquer tipo de objeto) ou MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT ou MQOO_SET (para filas) também deve ser especificado. A verificação executada é como para as outras opções especificadas, usando-se o identificador de usuário alternativo fornecido para a autoridade de objeto com nome específico, e a autoridade de aplicativo atual para a verificação MQZAO_ALTERNATE_USER_IDENTIFIER.
10. Essa autorização permite que qualquer *AlternateUserId* seja especificado.
11. Uma verificação MQZAO_OUTPUT também será executada se a fila não tiver um atributo de fila *Usage* de MQUS_TRANSMISSION.
12. A verificação executada é como para as outras opções especificadas, usando-se o identificador de usuário alternativo fornecido para a autoridade de fila com nome específico, e a autoridade de aplicativo atual para a verificação MQZAO_ALTERNATE_USER_IDENTIFIER.
13. A verificação será realizada somente se ambas as instruções a seguir forem verdadeiras:
    - Uma fila dinâmica permanente está sendo fechada e excluída.

- A fila não foi criada pela chamada MQOPEN que retornou a manipulação de objetos que estava sendo usada.

Caso contrário, não haverá verificação.

ALW *Autorizações para Comandos MQSC em PCFs Escape*

Estas informações resumem as autorizações necessárias para cada comando MQSC contido no PCF Escape.

Não aplicável significa que esta operação não é relevante para esse tipo de objeto.

O ID de usuário com o qual o programa que envia o comando está sendo executado também deve ter as seguintes autoridades:

- Autoridade MQZAO_CONNECT para o gerenciador de filas
- Autoridade MQZAO_DISPLAY no gerenciador de filas para executar comandos de PCF
- Autoridade para emitir o comando MQSC dentro do texto do comando PCF Escape

ALTER *object*

Object	**Authorization required**
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	MQZAO_CHANGE
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Listener	MQZAO_CHANGE
Serviço	MQZAO_CHANGE
Informações de comunicação	MQZAO_CHANGE

CLEAR *object*

Object	**Authorization required**
Fila	MQZAO_CLEAR
Tópico	MQZAO_CLEAR
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	Não-aplicável
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

Object	Authorization required
Informações de comunicação	Não-aplicável

DEFINE *object* NOREPLACE (“1” na página 145)

Object	Authorization required
Fila	MQZAO_CREATE **(“2” na página 145)**
Tópico	MQZAO_CREATE **(“2” na página 145)**
Processo	MQZAO_CREATE **(“2” na página 145)**
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CREATE **(“2” na página 145)**
Informações sobre Autenticação	MQZAO_CREATE **(“2” na página 145)**
Canal	MQZAO_CREATE **(“2” na página 145)**
Canal de conexão do cliente	MQZAO_CREATE **(“2” na página 145)**
Listener	MQZAO_CREATE **(“2” na página 145)**
Serviço	MQZAO_CREATE **(“2” na página 145)**
Informações de comunicação	MQZAO_CREATE **(“2” na página 145)**

DEFINE *object* REPLACE (“1” na página 145, “3” na página 145)

Object	Authorization required
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Listener	MQZAO_CHANGE
Serviço	MQZAO_CHANGE
Informações de comunicação	MQZAO_CHANGE

EXCLUIR *object*

Object	Authorization required
Fila	MQZAO_DELETE
Tópico	MQZAO_DELETE
Processo	MQZAO_DELETE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_DELETE

Object	Authorization required
Informações sobre Autenticação	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de conexão do cliente	MQZAO_DELETE
Listener	MQZAO_DELETE
Serviço	MQZAO_DELETE
Informações de comunicação	MQZAO_DELETE

DISPLAY *object*

Object	Authorization required
Fila	MQZAO_DISPLAY
Tópico	MQZAO_DISPLAY
Processo	MQZAO_DISPLAY
Gerenciador de filas	MQZAO_DISPLAY
Lista de Nomes	MQZAO_DISPLAY
Informações sobre Autenticação	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de conexão do cliente	MQZAO_DISPLAY
Listener	MQZAO_DISPLAY
Serviço	MQZAO_DISPLAY
Informações de comunicação	MQZAO_DISPLAY

START *object*

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Listener	MQZAO_CONTROL
Serviço	MQZAO_CONTROL
Informações de comunicação	Não-aplicável

PARAR *object*

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Listener	MQZAO_CONTROL
Serviço	MQZAO_CONTROL
Informações de comunicação	Não-aplicável

Comandos do Canal

Comando:	Object	Authorization required
PING CHANNEL	Canal	MQZAO_CONTROL
RESET CHANNEL	Canal	MQZAO_CONTROL_EXTENDED
RESOLVE CHANNEL	Canal	MQZAO_CONTROL_EXTENDED

Comandos de Assinatura

Comando:	Object	Authorization required
ALTER SUB	Tópico	MQZAO_CONTROL
DEFINE SUB	Tópico	MQZAO_CONTROL
DELETE SUB	Tópico	MQZAO_CONTROL
DISPLAY SUB	Tópico	MQZAO_DISPLAY

Comandos de Segurança

Comando:	Object	Authorization required
SET AUTHREC	Gerenciador de filas	MQZAO_CHANGE
DELETE AUTHREC	Gerenciador de filas	MQZAO_CHANGE
DISPLAY AUTHREC	Gerenciador de filas	MQZAO_DISPLAY
DISPLAY AUTHSERV	Gerenciador de filas	MQZAO_DISPLAY
DISPLAY ENTAUTH	Gerenciador de filas	MQZAO_DISPLAY
SET CHLAUTH	Gerenciador de filas	MQZAO_CHANGE
DISPLAY CHLAUTH	Gerenciador de filas	MQZAO_DISPLAY
REFRESH SECURITY	Gerenciador de filas	MQZAO_CHANGE

Exibições de status

Comando:	Object	Authorization required
DISPLAY CHSTATUS	Gerenciador de filas	MQZAO_DISPLAY Observe que a autoridade `+inq` (ou equivalentemente MQZAO_INQUIRE) será necessária na fila de transmissão se o tipo de canal for CLUSSDR.
DISPLAY LSSTATUS	Gerenciador de filas	MQZAO_DISPLAY
DISPLAY PUBSUB	Gerenciador de filas	MQZAO_DISPLAY
DISPLAY SBSTATUS	Gerenciador de filas	MQZAO_DISPLAY
DISPLAY SVSTATUS	Gerenciador de filas	MQZAO_DISPLAY
DISPLAY TPSTATUS	Gerenciador de filas	MQZAO_DISPLAY

Comandos do Cluster

Comando:	Object	Authorization required
EXIBIR CLUSQMGR	Gerenciador de filas	MQZAO_DISPLAY
REFRESH CLUSTER	associação ao grupo 'mqm' necessária	
RESET CLUSTER	associação ao grupo 'mqm' necessária	
SUSPEND QMGR	associação ao grupo 'mqm' necessária	
RESUME QMGR	associação ao grupo 'mqm' necessária	

Outros comandos administrativos

Comando:	Object	Authorization required
PING QMGR	Gerenciador de filas	MQZAO_DISPLAY
REFRESH QMGR	Gerenciador de filas	MQZAO_CHANGE
RESET QMGR	Gerenciador de filas	MQZAO_CHANGE
DISPLAY CONN	Gerenciador de filas	MQZAO_DISPLAY
STOP CONN	Gerenciador de filas	MQZAO_CHANGE

Nota:

1. Para comandos DEFINE, a autoridade MQZAO_DISPLAY também será necessária para o objeto LIKE, se houver um especificado, ou no objeto SYSTEM.DEFAULT.xxx apropriado, se LIKE for omitido.
2. A autoridade MQZAO_CREATE não é específica a um determinado objeto ou tipo de objeto. A autoridade Criar é concedida para todos os objetos de um gerenciador de filas especificado, especificando-se um tipo de objeto de QMGR no comando `setmqaut`.
3. Isso se aplicará se o objeto a ser substituído já existir. Caso contrário, a verificação será como para DEFINE *object* NOREPLACE.

Informações relacionadas

Armazenamento em Cluster: Usando Melhores Práticas de REFRESH CLUSTER

ALW *Autorizações para Comandos PCF*

Esta seção resume as autorizações necessárias para cada comando PCF.

Nenhuma verificação significa que nenhuma verificação de autorização é executada; *Não aplicável* significa que esta operação não é relevante para este tipo de objeto.

O ID de usuário com o qual o programa que envia o comando está sendo executado também deve ter as seguintes autoridades:

- Autoridade MQZAO_CONNECT para o gerenciador de filas
- Autoridade MQZAO_DISPLAY no gerenciador de filas para executar comandos de PCF

A autorização especial MQZAO_ALL_ADMIN inclui todas as autorizações na lista a seguir que são relevantes ao tipo de objeto, exceto MQZAO_CREATE, que não é específica a um determinado objeto ou tipo de objeto.

Mudar *object*

Object	**Authorization required**
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de Filas	MQZAO_CHANGE
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Receptor	MQZAO_CHANGE
Serviço	MQZAO_CHANGE
Informações de Comunicação	MQZAO_CHANGE

Clear *object*

Object	**Authorization required**
Fila	MQZAO_CLEAR
Tópico	MQZAO_CLEAR
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	Não-aplicável
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável
Informações de comunicação	Não-aplicável

Copiar *object* (sem substituir) (1)

Object	Authorization required
Fila	MQZAO_CREATE **(2)**
Tópico	MQZAO_CREATE **(2)**
Processo	MQZAO_CREATE **(2)**
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CREATE **(2)**
Informações sobre Autenticação	MQZAO_CREATE **(2)**
Canal	MQZAO_CREATE **(2)**
Canal de conexão do cliente	MQZAO_CREATE **(2)**
Receptor	MQZAO_CREATE **(2)**
Serviço	MQZAO_CREATE **(2)**
Informações de Comunicação	MQZAO_CREATE **(“2” na página 152)**

Copie o *object* (com substituição) (1, 4)

Object	Authorization required
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Receptor	MQZAO_CHANGE
Serviço	MQZAO_CHANGE
Informações de Comunicação	MQZAO_CHANGE

Criar *object* (sem substituir) (3)

Object	Authorization required
Fila	MQZAO_CREATE **(2)**
Tópico	MQZAO_CREATE **(2)**
Processo	MQZAO_CREATE **(2)**
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CREATE **(2)**
Informações sobre Autenticação	MQZAO_CREATE **(2)**
Canal	MQZAO_CREATE **(2)**

Object	Authorization required
Canal de conexão do cliente	MQZAO_CREATE **(2)**
Receptor	MQZAO_CREATE **(2)**
Serviço	MQZAO_CREATE **(2)**
Informações de Comunicação	MQZAO_CREATE **(2)**

Crie o *object* (com substituição) (3, 4)

Object	Authorization required
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Receptor	MQZAO_CHANGE
Serviço	MQZAO_CHANGE
Informações de Comunicação	MQZAO_CHANGE

Excluir *object*

Object	Authorization required
Fila	MQZAO_DELETE
Tópico	MQZAO_DELETE
Processo	MQZAO_DELETE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_DELETE
Informações sobre Autenticação	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de conexão do cliente	MQZAO_DELETE
Receptor	MQZAO_DELETE
Serviço	MQZAO_DELETE
Informações de Comunicação	MQZAO_DELETE

Inquire *object*

Object	Authorization required
Fila	MQZAO_DISPLAY
Tópico	MQZAO_DISPLAY

Object	Authorization required
Processo	MQZAO_DISPLAY
Gerenciador de Filas	MQZAO_DISPLAY
Lista de Nomes	MQZAO_DISPLAY
Informações sobre Autenticação	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de conexão do cliente	MQZAO_DISPLAY
Receptor	MQZAO_DISPLAY
Serviço	MQZAO_DISPLAY
Informações de Comunicação	MQZAO_DISPLAY

Inquire *object* names

Object	Authorization required
Fila	Sem verificação
Tópico	Sem verificação
Processo	Sem verificação
Gerenciador de filas	Sem verificação
Lista de Nomes	Sem verificação
Informações sobre Autenticação	Sem verificação
Canal	Sem verificação
Canal de conexão do cliente	Sem verificação
Listener	Sem verificação
Serviço	Sem verificação
Informações de comunicação	Sem verificação

Iniciar *object*

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Receptor	MQZAO_CONTROL
Serviço	MQZAO_CONTROL

Object	Authorization required
Informações de comunicação	Não-aplicável

Parar *object*

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Receptor	MQZAO_CONTROL
Serviço	MQZAO_CONTROL
Informações de comunicação	Não-aplicável

Comandos do Canal

Comando:	Object	Authorization required
Executar ping no Canal	Canal	MQZAO_CONTROL
Redefinir Canal	Canal	MQZAO_CONTROL_EXTENDED
Resolver Canal	Canal	MQZAO_CONTROL_EXTENDED

Comandos de Assinatura

Comando:	Object	Authorization required
Mudar assinatura	Tópico	MQZAO_CONTROL
Criar assinatura	Tópico	MQZAO_CONTROL
Excluir assinatura	Tópico	MQZAO_CONTROL
Consultar Assinatura	Tópico	MQZAO_DISPLAY

Comandos de Segurança

Comando:	Object	Authorization required
Configurar Registro de Autoridade	Gerenciador de filas	MQZAO_CHANGE
Excluir Registro de Autoridade	Gerenciador de filas	MQZAO_CHANGE
Consultar Registros de Autoridade	Gerenciador de filas	MQZAO_DISPLAY
Consultar Autoridade de Serviço	Gerenciador de filas	MQZAO_DISPLAY
Solicitar Autoridade de Entidade	Gerenciador de filas	MQZAO_DISPLAY

Comando:	Object	Authorization required
Configurar Registro de Autenticação de Canal	Gerenciador de filas	MQZAO_CHANGE
Solicitar Registros de Autenticação de Canal	Gerenciador de filas	MQZAO_DISPLAY
Atualizar segurança	Gerenciador de filas	MQZAO_CHANGE

Exibições de status

Comando:	Object	Authorization required
Consultar Status do Canal	Gerenciador de filas	MQZAO_DISPLAY Observe que a autoridade `+inq` (ou equivalentemente MQZAO_INQUIRE) será necessária na fila de transmissão se o tipo de canal for CLUSSDR.
Consulte o status do ouvinte de canal	Gerenciador de filas	MQZAO_DISPLAY
Investigar Status da Pub/Ass	Gerenciador de filas	MQZAO_DISPLAY
Consultar status da assinatura	Gerenciador de filas	MQZAO_DISPLAY
Consultar Status do Serviço	Gerenciador de filas	MQZAO_DISPLAY
Consultar status de tópico	Gerenciador de filas	MQZAO_DISPLAY

Comandos do Cluster

Comando:	Object	Authorization required
Consultar Gerenciador de Filas de Clusters	Gerenciador de filas	MQZAO_DISPLAY
Refresh Cluster	associação ao grupo 'mqm' necessária	associação ao grupo 'mqm' necessária
Reset Cluster	associação ao grupo 'mqm' necessária	associação ao grupo 'mqm' necessária
Suspender Cluster de Gerenciador de Filas	associação ao grupo 'mqm' necessária	associação ao grupo 'mqm' necessária
Retomar Cluster de Gerenciador de Filas	associação ao grupo 'mqm' necessária	associação ao grupo 'mqm' necessária

Outros comandos administrativos

Comando:	Object	Authorization required
Executar Ping do Gerenciador de Filas	Gerenciador de filas	MQZAO_DISPLAY
Atualizar Gerenciador de Filas	Gerenciador de filas	MQZAO_CHANGE
Reconfigurar Gerenciador de Filas	Gerenciador de filas	MQZAO_CHANGE

Comando:	Object	Authorization required
Reconfigurar as Estatísticas de Fila	Fila	MQZAO_DISPLAY e MQZAO_CHANGE
Consultar Conexão	Gerenciador de filas	MQZAO_DISPLAY
Para Conexão	Gerenciador de filas	MQZAO_CHANGE

Nota:

1. Para comandos Copy, a autoridade MQZAO_DISPLAY também é necessária para o objeto De.
2. A autoridade MQZAO_CREATE não é específica a um determinado objeto ou tipo de objeto. A autoridade Criar é concedida para todos os objetos de um gerenciador de filas especificado, especificando-se um tipo de objeto de QMGR no comando `setmqaut`.
3. Para comandos de Criação, a autoridade MQZAO_DISPLAY também é necessária para o SYSTEM.DEFAULT.* objeto.
4. Isso se aplicará se o objeto a ser substituído já existir. Caso contrário, a verificação será como para Copy ou Create sem substituição.

AIX Criando e gerenciando grupos no AIX

No AIX, desde que você não esteja usando NIS ou NIS+, use SMITTY para trabalhar com grupos.

Sobre esta tarefa

No AIX, é possível usar SMITTY para criar um grupo, incluir um usuário em um grupo, exibir uma lista dos usuários que estão no grupo e remover um usuário de um grupo.

Procedimento

1. No SMITTY, selecione **Segurança e usuários** e pressione Enter.
2. Selecione **Grupos** e pressione Enter.
3. Para criar um grupo, conclua as etapas a seguir:
   a) Selecione **Incluir um grupo** e pressione Enter.
   b) Insira o nome do grupo e os nomes de usuários que você deseja incluir no grupo, separados por vírgulas.
   c) Pressione Enter para criar o grupo.
4. Para incluir um usuário em um grupo, conclua as etapas a seguir:
   a) Selecione **Mudar/mostrar características de grupos** e pressione Enter.
   b) Insira o nome do grupo para mostrar uma lista dos membros do grupo.
   c) Inclua os nomes dos usuários que você deseja incluir no grupo, separados por vírgulas.
   d) Pressione Enter para incluir os nomes no grupo.
5. Para exibir quem está em um grupo, conclua as etapas a seguir:
   a) Selecione **Mudar/mostrar características de grupos** e pressione Enter.
   b) Insira o nome do grupo para mostrar uma lista dos membros do grupo.
6. Para remover um usuário de um grupo, conclua as etapas a seguir:
   a) Selecione **Mudar/mostrar características de grupos** e pressione Enter.
   b) Insira o nome do grupo para mostrar uma lista dos membros do grupo.
   c) Exclua o nome do usuário que você deseja remover do grupo.
   d) Pressione Enter para remover o nome do grupo.

Linux Criando e gerenciando grupos no Linux

No Linux, desde que você não esteja usando NIS ou NIS+, use o arquivo `/etc/group` para trabalhar com grupos.

Sobre esta tarefa

No Linux, as informações do grupo são mantidas no arquivo do `/etc/group`. É possível usar comandos para criar um grupo, incluir um usuário em um grupo, exibir uma lista dos usuários que estão no grupo e remover um usuário de um grupo.

Procedimento

1. Para criar um novo grupo, use o comando **groupadd**.
   Digite o seguinte comando:
   ```
   groupadd -g group-ID group-name
   ```
   em que *group-ID* é o identificador numérico do grupo e *group-name* é o nome do grupo.
2. Para incluir um membro em um grupo complementar, use o comando **usermod** para listar os grupos complementares dos quais o utilizador é actualmente membro, e os grupos complementares que o usuário deve se tornar um membro do.
   Por exemplo, se o usuário já é um membro do grupo `groupa` e deve se tornar um membro de `groupb`, use o comando a seguir:
   ```
   usermod -G groupa,groupb user-name
   ```
   em que *user-name* é o nome do usuário.
3. Para exibir quem é um membro de um grupo, use o comando **getent**.
   Digite o seguinte comando:
   ```
   getent group group-name
   ```
   em que *group-name* é o nome do grupo.
4. Para remover um membro de um grupo complementar, use o comando **usermod** para listar os grupos complementares dos quais você deseja que o usuário permaneça um membro.
   Por exemplo, se o grupo primário do usuário for `users` e o usuário também for um membro dos grupos `mqm`, `groupa` e `groupb`, para remover o usuário do grupo `mqm`, use o comando a seguir:
   ```
   usermod -G groupa,groupb user-name
   ```
   em que *user-name* é o nome do usuário.

Windows Criando e gerenciando grupos no Windows

No Windows, use o recurso Gerenciamento de computadores para administrar grupos em uma estação de trabalho ou máquina do servidor de membro.

Sobre esta tarefa

Para controladores de domínio, usuários e grupos são administrados por meio do Active Directory. Para obter mais detalhes sobre como usar o Active Directory, consulte as instruções apropriadas do sistema operacional.

Todas as mudanças feitas na associação ao grupo de um proprietário não são reconhecidas até que o gerenciador de filas seja reiniciado ou você emita o comando MQSC **REFRESH SECURITY** (ou o equivalente PCF).

Use o painel Gerenciamento de computadores do Windows para trabalhar com o usuário e os grupos. Quaisquer mudanças feitas no usuário com login efetuado atual podem não entrar em vigor até que o usuário efetue login novamente.

Windows *Criando um grupo no Windows*

Crie um grupo usando o painel de controle.

Procedimento

1. Abra o painel de controle
2. Dê um clique duplo em **Ferramentas Administrativas**.
   O painel Ferramentas Administrativas é aberto.
3. Dê um clique duplo em **Gerenciamento de Computadores**.
   O painel Gerenciamento de Computadores é aberto.
4. Expanda **Local Users and Groups**.
5. Clique com o botão direito do mouse em **Grupos** e selecione **Novo Grupo...**.
   O painel Novo Grupo é exibido.
6. Digite um nome apropriado no campo de nome Grupo e, em seguida, clique em **Criar**.
7. Clique em **Fechar**.

Windows *Incluindo um usuário em um grupo no Windows*

Inclua um usuário em um grupo usando o painel de controle.

Procedimento

1. Abra o painel de controle
2. Dê um clique duplo em **Ferramentas Administrativas**.
   O painel Ferramentas Administrativas é aberto.
3. Dê um clique duplo em **Gerenciamento de Computadores**.
   O painel Gerenciamento de Computadores é aberto.
4. No painel Gerenciamento de Computadores, expanda **Usuários e Grupos Locais**.
5. Selecione **Usuários**
6. Clique duas vezes no usuário que você deseja incluir em um grupo.
   O painel de propriedades do usuário é exibido.
7. Selecione a guia **Membro de**.
8. Selecione o grupo no qual você deseja incluir o usuário. Se o grupo desejado não estiver visível:
   a) Clique em **Incluir...**.
      O painel Selecionar Grupos é exibido.
   b) Clique em **Locais...**.
      O painel Locais é exibido.
   c) Selecione o local do grupo em que você deseja incluir o usuário na lista e clique em **OK**.
   d) Digite o nome do grupo no campo fornecido.

      Como alternativa, clique em **Avançado ...** .. e, em seguida, **Localizar Agora** para listar os grupos disponíveis no local atualmente selecionado Daqui, selecione o grupo em que deseja incluir o usuário e clique em **OK**.
   e) Clique em **OK**.
      O painel de propriedades do usuário é exibido, mostrando o grupo incluído.
   f) Selecionar o grupo.
9. Clique em **OK**.

O painel Gerenciamento de Computadores é exibido.

Windows *Exibindo quem está em um grupo em Windows*

Exiba os membros de um grupo usando o painel de controle.

Procedimento

1. Abra o painel de controle
2. Dê um clique duplo em **Ferramentas Administrativas**.
   O painel Ferramentas Administrativas é aberto.
3. Dê um clique duplo em **Gerenciamento de Computadores**.
   O painel Gerenciamento de Computadores é aberto.
4. No painel Gerenciamento de Computadores, expanda **Usuários e Grupos Locais**.
5. Selecione **Grupos**.
6. Clique duas vezes em um grupo. O painel de propriedades do grupo é exibido.
   O painel de propriedades do grupo é exibido.

Resultados

Os membros do grupo são exibidos.

Windows *Removendo um usuário de um grupo em Windows*

Remova um usuário de um grupo usando o painel de controle.

Procedimento

1. Abra o painel de controle
2. Dê um clique duplo em **Ferramentas Administrativas**.
   O painel Ferramentas Administrativas é aberto.
3. Dê um clique duplo em **Gerenciamento de Computadores**.
   O painel Gerenciamento de Computadores é aberto.
4. No painel Gerenciamento de Computadores, expanda **Usuários e Grupos Locais**.
5. Selecione **Usuários**.
6. Clique duas vezes no usuário que você deseja incluir em um grupo.
   O painel de propriedades do usuário é exibido.
7. Selecione a guia **Membro de**.
8. Selecione o grupo do qual você deseja remover o usuário e, em seguida, clique em **Remover**.
9. Clique em **OK**.
   O painel Gerenciamento de Computadores é exibido.

Resultados

Agora você removeu o usuário do grupo.

Windows Considerações especiais para segurança no Windows

Algumas funções de segurança se comportam de forma diferente em diferentes versões do Windows.

A segurança do IBM MQ depende de chamadas para a API do sistema operacional para obter informações sobre autorizações de usuários e associações de grupo. Algumas funções não se comportam de modo idêntico em sistemas Windows. Esta coleção de tópicos inclui descrições de como essas diferenças podem afetar a segurança do IBM MQ quando se está executando o IBM MQ em um ambiente do Windows.

Windows *Contas de usuário local e de domínio para o serviço IBM MQ Windows*

Quando o IBM MQ estiver em execução, ele deverá verificar se apenas usuários autorizados podem acessar gerenciadores de filas ou filas. Isso requer uma conta de usuário especial que o IBM MQ pode usar para consultar informações sobre o usuário que está tentando tal acesso.

- "Configurando contas de usuário especiais com o Prepare IBM MQ Wizard" na página 156
- "Usando o IBM MQ com o Active Directory" na página 156
- "Direitos de usuário necessários para um serviço do IBM MQ Windows" na página 157

Configurando contas de usuário especiais com o Prepare IBM MQ Wizard

O Prepare IBM MQ Wizard cria uma conta do usuário especial para que o serviço Windows possa ser compartilhado por processos que precisam usá-lo (consulte Configurando o IBM MQ com o PPrepare IBM MQ Wizard).

Um serviço é compartilhado entre os processos do cliente Windows para uma instalação do IBM MQ. Um serviço é criado para cada instalação. Cada serviço é denominado `MQ_`*`InstallationName`*e possui um nome de exibição `IBM MQ(`*`InstallationName`*`)`.

Como cada serviço deve ser compartilhado entre as sessões de logon não interativas e interativas, deve-se ativar cada uma delas sob uma conta de usuário especial. É possível usar uma conta do usuário especial para todos os serviços ou criar contas do usuário especiais diferentes. Cada conta de usuário especial deve ter o direito de usuário para `Efetuar logon como um serviço`; para obter mais informações, consulte Tabela 14 na página 157. Se o ID do usuário não tiver a autoridade para executar o serviço, o serviço não será iniciado e retornará um erro no log de eventos do sistema Windows. Geralmente, você terá executado o Prepare IBM MQ Wizard e configurará o ID do usuário corretamente. No entanto, se você configurou o ID do usuário manualmente, possivelmente ocorrerá um problema que precisará ser resolvido.

Quando você instala o IBM MQ e executa o Prepare IBM MQ Wizard pela primeira vez, ele cria uma conta do usuário local para o serviço chamado MUSR_MQADMIN com as configurações e permissões necessárias, incluindo `Efetuar logon como um serviço`.

Para instalações subsequentes, o Prepare IBM MQ Wizard cria uma conta do usuário denominada MUSR_MQADMIN*x*, em que *x* é o próximo número disponível que representa um ID do usuário que não existe. A senha para MUSR_MQADMIN*x* é gerada aleatoriamente quando a conta é criada e usada para configurar o ambiente de logon para o serviço. A senha gerada não expira.

Esta conta do IBM MQ não é afetada por quaisquer políticas de conta que são configuradas no sistema para exigir que as senhas das contas sejam mudadas após um determinado período.

A senha não é conhecida fora desse processamento único e é armazenada pelo sistema operacional Windows em uma parte segura do registro.

Usando o IBM MQ com o Active Directory

Em algumas configurações de rede, em que as contas do usuário são definidas nos controladores de domínio que estão utilizando o serviço de diretório do Active Directory, a conta do usuário local sob a qual o IBM MQ está sendo executado pode não ter a autoridade necessária para consultar a associação ao grupo de outras contas de usuário de domínio. Quando você instala o IBM MQ, o Prepare IBM MQ Wizard identifica se esse é o caso executando testes e fazendo perguntas sobre a configuração de rede.

Se a conta do usuário local que o IBM MQ está executando sob não possui a autoridade necessária, o Prepare IBM MQ Wizard solicitará os detalhes da conta de um domínio conta do usuário com direitos de usuário específicos. Para obter informações sobre como criar e configurar uma conta de domínio do Windows, consulte Criando e configurando contas de domínio do Windows para o IBM MQ. Para obter os direitos de usuário que a conta do usuário do domínio requer, consulte Tabela 14 na página 157.

Quando você tiver inserido detalhes da conta válidos para a conta do usuário do domínio no Prepare IBM MQ Wizard, o assistente configurará um serviço IBM MQ Windows para ser executado na nova conta. Os detalhes da conta são retidos na parte segura do Registro e não podem ser lidos pelos usuários.

Quando o serviço está em execução, um serviço do IBM MQ Windows é iniciado e permanece em execução enquanto o serviço estiver em execução. Um administrador do IBM MQ que efetua logon no servidor depois que o serviço do Windows é ativado pode usar o IBM MQ Explorer para administrar gerenciadores de filas no servidor. Ele conecta o IBM MQ Explorer ao processo do serviço do Windows existente. Essas duas ações precisam de diferentes níveis de permissão para que possam trabalhar:

- O processo de ativação requer uma permissão de ativação.
- O administrador do IBM MQ requer permissão de acesso.

Direitos de usuário necessários para um serviço do IBM MQ Windows

A tabela a seguir lista os direitos de usuário necessários para as contas de usuário local e de domínio sob as quais o serviço do Windows para uma instalação do IBM MQ é executado.

Tabela 14. Direitos de usuário necessários para um serviço do Windows do IBM MQ

Permissão	**Descrição**
Efetuar logon como uma tarefa em lote	Permite que um serviço do IBM MQ Windows execute sob essa conta do usuário.
Efetue logon como serviço	Permite que os usuários configurem o serviço do IBM MQ Windows para efetuar logon utilizando a conta configurada.
Desligar o sistema	Permite que o serviço IBM MQ Windows reinicie o servidor, se estiver configurado para fazer isso quando a recuperação de um serviço falhar.
Aumentar Cotas	Necessário para chamada de `CreateProcessAsUser` do sistema operacional.
Aja como parte do sistema operacional	Requerido para a chamada de `LogonUser` do sistema operacional.
Verificação de passagem de desvio	Requerido para a chamada de `LogonUser` do sistema operacional.
Substituir um símbolo em nível de processo	Requerido para a chamada de `LogonUser` do sistema operacional.

Nota: Podem ser necessários direitos de programas de depuração em ambientes que executam os aplicativos ASP e IIS.

Sua conta de usuário de domínio deve ter esses direitos de usuário do Windows configurados como direitos de usuário efetivo, conforme listado no aplicativo Política de Segurança Local. Se eles não forem, configure-os usando o aplicativo Política de Segurança Local localmente no servidor ou usando o domínio de Aplicativo de Segurança do Domínio amplo.

Windows *Permissões de segurança do Windows Server*

A instalação do IBM MQ se comporta de forma diferente no Windows Server, dependendo se um usuário local ou um usuário do domínio executa a instalação.

Se um usuário *local* instala o IBM MQ, o Prepare IBM MQ Wizard detecta que o usuário local criado para o serviço do IBM MQ Windows pode recuperar as informações de associação ao grupo do usuário de instalação. O Prepare IBM MQ Wizard faz perguntas ao usuário sobre a configuração de rede para determinar se há outras contas do usuário definidas em controladores de domínio em execução no Windows 2000 ou mais recente. Se sim, o serviço do IBM MQ Windows precisar ser executado em uma conta de usuário do domínio com configurações e autoridades específicas. O Prepare IBM MQ Wizard solicita ao usuário os detalhes da conta desse usuário conforme descrito em Configurando o IBM MQ com o Prepare IBM MQ Wizard.

Se um usuário *domain* instala o IBM MQ, o Prepare IBM MQ Wizard detecta que o usuário local criado para o serviço do IBM MQ Windows não pode recuperar as informações de associação ao grupo do usuário de instalação. Neste caso, o Prepare IBM MQ Wizard sempre solicita ao usuário os detalhes da conta da conta do usuário do domínio para o serviço do IBM MQ Windows usar.

Quando o serviço do IBM MQ Windows precisa usar uma conta de usuário de domínio, o IBM MQ não pode operar corretamente até que isso tenha sido configurado usando o Prepare IBM MQ Wizard. O Prepare IBM MQ Wizard não permite que o usuário continue com outras tarefas, até que o serviço do Windows tenha sido configurado com uma conta adequada.

Para obter mais informações, consulte Criando e configurando contas de domínio para o IBM MQ.

Windows *Mudando o nome do usuário associado ao serviço do IBM MQ*

É possível mudar o nome do usuário associado ao serviço do IBM MQ criando uma nova conta e inserindo seus detalhes usando o Prepare IBM MQ Wizard.

Sobre esta tarefa

Quando você instala o IBM MQ e executa o Prepare IBM MQ Wizard pela primeira vez, ele cria uma conta do usuário local para o serviço chamado MUSR_MQADMIN. Para instalações subsequentes, o Prepare IBM MQ Wizard cria uma conta do usuário denominada MUSR_MQADMIN*x*, em que *x* é o próximo número disponível que representa um ID do usuário que não existe.

Pode ser necessário mudar o nome do usuário associado ao serviço do IBM MQ de MUSR_MQADMIN ou MUSR_MQADMIN*x* para algo diferente. Por exemplo, talvez você precise fazer isso se seu gerenciador de filas estiver associado ao Db2, que não aceita nomes de usuário com mais de 8 caracteres.

Procedimento

1. Crie uma nova conta de usuário (por exemplo **NEW_NAME**)
2. Use o Prepare IBM MQ Wizard para inserir os detalhes do nova conta do usuário.

Tarefas relacionadas

Configurando o IBM MQ com o Prepare IBM MQ Wizard

Windows *Mudando a senha da conta do usuário local do serviço IBM MQ Windows*

É possível mudar a senha da conta do usuário local do serviço IBM MQ Windows usando o painel Gerenciamento de computadores.

Sobre esta tarefa

Para mudar a senha do IBM MQ Windows de serviço da conta de usuário local, execute as seguintes etapas:

Procedimento

1. Identifique o usuário do serviço está sendo executado.
2. Pare o serviço do IBM MQ a partir do painel do Computer Management.
3. Mude a senha necessária da mesma maneira que você mudaria a senha de um indivíduo.
4. Acesse as propriedades para o serviço do IBM MQ a partir do painel do Computer Management.
5. Selecione a página **Efetuar Logon** .
6. Confirme se o nome da conta especificado corresponde ao usuário para o qual a senha foi modificada.
7. Digite a senha no **Senha** e **Confirmar Senha** os campos e clique em **OK**.

Windows *Mudando a senha para um serviço IBM MQ Windows para uma instalação em execução em uma conta de usuário do domínio*

Como uma alternativa ao uso do Prepare IBM MQ Wizard para inserir os detalhes da conta da conta do usuário do domínio, é possível usar o painel Gerenciamento do Computador para alterar os detalhes de **Logon** para o Serviço IBM MQ específico da instalação.

Sobre esta tarefa

Se o serviço IBM MQ Windows para uma instalação estiver em execução em uma conta de usuário do domínio, você poderá mudar a senha para a conta conforme a seguir:

Procedimento

1. Mude a senha para a conta de domínio no controlador de domínio. Talvez seja necessário solicitar ao seu administrador de domínio para fazer isso para você.
2. Conclua as etapas a seguir para modificar a página **Efetuar logon** para o serviço do IBM MQ.
   a) Identifique o usuário sob o qual o serviço está sendo executado.
   b) Pare o serviço do IBM MQ a partir do painel do Computer Management.
   c) Mude a senha necessária da mesma maneira que você mudaria a senha de um indivíduo.
   d) Acesse as propriedades para o serviço do IBM MQ a partir do painel do Computer Management.
   e) Selecione a página **Efetuar Logon** .
   f) Confirme se o nome da conta especificado corresponde ao usuário para o qual a senha foi modificada.
   g) Digite a senha no **Senha** e **Confirmar Senha** os campos e clique em **OK**.

   A conta do usuário sob a qual executa o serviço IBM MQ Windows executa quaisquer comandos MQSC que são emitidos por aplicativos de interface com o usuário ou executados automaticamente na inicialização, encerramento ou recuperação de serviço do sistema. Essa conta do usuário deve, portanto, ter direitos de administração do IBM MQ. Por padrão, ele é incluído no grupo mqm local no servidor. Se essa associação é removida, o serviço do IBM MQ Windows não funciona. Para obter mais informações sobre os direitos do usuário, consulte “Direitos de usuário necessários para um serviço do IBM MQ Windows” na página 157.

   Se um problema de segurança surgir com a conta do usuário sob a qual o serviço do IBM MQ Windows executa, mensagens de erro e descrições aparecem no log de eventos do sistema.

Tarefas relacionadas

Configurando o IBM MQ com o Prepare IBM MQ Wizard

Windows ***Considerações ao promover servidores Windows para controladores de domínio***

Ao promover um servidor Windows para um domínio, é necessário considerar se a configuração de segurança relacionada às permissões de usuário e de grupo é apropriada. Ao mudar o estado de uma máquina do Windows entre o servidor e o controlador de domínio, é necessário levar em consideração que isso pode afetar a operação do IBM MQ porque o IBM MQ usa um grupo mqm definido localmente.

Configurações de segurança relacionadas a permissões de usuário e grupo de domínio

O IBM MQ conta com informações de associação ao grupo para implementar sua política de segurança, o que significa que é importante que o ID do usuário que está executando operações do IBM MQ possa determinar as associações ao grupo de outros usuários.

Ao promover um servidor Windows para um controlador de domínio, é apresentada uma opção para a configuração de segurança relacionada a permissões de usuário e de grupo. Essa opção controla se usuários arbitrários poderão recuperar associações do grupo do Active Directory. Se um controlador

de domínio é configurado para que as contas locais tenham a autoridade para consultar a associação ao grupo das contas de usuário de domínio, o ID do usuário padrão criado pelo IBM MQ durante o processo de instalação pode obter associações de grupo para outros usuários, conforme necessário. No entanto, se um controlador de domínio for configurado para que as contas locais não tenham a autoridade para consultar a associação ao grupo das contas de usuário de domínio, isso impedirá que o IBM MQ conclua suas verificações de que os usuários que estão definidos no domínio estão autorizados a acessar gerenciadores de filas ou filas e o acesso falhará. Se você estiver usando o Windows em um controlador de domínio que tenha tiver sido configurado desta forma, uma conta de usuário do domínio especial com as permissões necessárias deve ser usada.

Neste caso, você precisa saber:

- Como as permissões de segurança para sua versão do Windows comporte-se.
- Como permitir que membros do grupo mqm de domínio leiam associação ao grupo.
- Como configurar um serviço do IBM MQ Windows para ser executado sob um usuário do domínio.

Para obter mais informações, consulte Configurando contas do usuário para o IBM MQ.

Acesso doIBM MQ para o grupo mqm local

Quando os servidores Windows são promovidos ou rebaixados para controladores de domínio, o IBM MQ perde o acesso ao grupo mqm local.

Quando um servidor é promovido para controlador de domínio, o escopo é alterado de local para domínio local. Quando a máquina é rebaixada para servidor, todos os grupos locais do domínio são removidos. Isso significa que alterar uma máquina de servidor para controlador de domínio e novamente para servidor perde o acesso a um grupo mqm local. O sintoma é um erro indicando a perda de um grupo mqm local, por exemplo:

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

Para resolver esse problema, recrie o grupo mqm local usando as ferramentas de gerenciamento padrão do Windows. Como todas as informações de associação ao grupo são perdidas, deve-se restabelecer os usuários privilegiados do IBM MQ no grupo mqm local recém criado. Se a máquina for um membro de domínio, também se deve incluir o grupo mqm de domínio no grupo mqm local, a fim de conceder aos IDs de usuário privilegiado de domínio do IBM MQ o nível de autoridade necessário.

Windows *Restrições em grupos aninhados no Windows*

Há restrições no uso de grupos aninhados. Estas resultam, em parte, do nível funcional do domínio e, em parte, das restrições do IBM MQ.

O Active Directory pode suportar diferentes tipos de grupos em um contexto de domínio, dependendo do nível funcional do domínio. Por padrão, os domínios Windows 2003 estão no nível funcional "Windows 2000 combinado". (Windows O Server 2008 e o Windows Server 2012 seguem o modelo de domínio Windows 2003.) O nível funcional do domínio determina os tipos de grupos suportados e o nível de aninhamento permitido ao configurar IDs do usuário e um ambiente de domínio. Consulte a documentação do Active Directory para obter detalhes sobre o Escopo de Grupo e os critérios de inclusão.

Além de requisitos de Active Directory, ainda se acrescentam restrições em IDs usados pelo IBM MQ. As APIs de rede usadas pelo IBM MQ não suportam todas as configurações que são suportadas pelo nível funcional de domínio. Como resultado, o IBM MQ não é capaz de consultar as associações do grupo de todos os IDs de Domínio presentes em um grupo de Domínio Local, que é então aninhado em um grupo local. Além disso, o aninhamento múltiplo de grupos globais e universais não é suportado. No entanto, grupos globais e universais imediatamente aninhados são suportados.

Windows *Autorizando Usuários a Usar o IBM MQ Remotamente*

Se você precisar criar e iniciar gerenciadores de filas quando conectado ao IBM MQ remotamente, deverá ter o acesso de usuário `Criar objetos globais`.

Sobre esta tarefa

Nota: Os administradores possuem o acesso de usuário `Create global objects` por padrão; portanto, se você for um administrador, será possível criar e iniciar gerenciadores de filas quando conectado remotamente, sem alterar seus direitos de usuário.

Se você estiver se conectando a uma máquina Windows usando os Serviços de terminal ou uma Conexão de área de trabalho remota e tiver problemas ao criar, iniciar ou excluir um gerenciador de filas, isso poderá ser porque você não tem o acesso de usuário `Criar objetos globais`.

O acesso de usuário `Create global objects` limita os usuários autorizados a criarem objetos no namespace global. Para que um aplicativo crie um objeto global, ele deve estar em execução no namespace global ou o usuário com o qual o aplicativo está sendo executado deve ter o acesso de usuário `Create global objects` aplicado a ele.

Quando você se conecta remotamente a uma máquina Windows utilizando os Serviços de terminal ou uma Conexão de área de trabalho remota, os aplicativos são executados em seus próprios namespaces locais. Se você tentar criar ou excluir um gerenciador de filas usando o IBM MQ Explorer ou o comando **crtmqm** ou **dltmqm**, ou iniciar um gerenciador de filas usando o comando **strmqm**, o resultado será uma falha de autorização. Isso cria um IBM MQ FDC com o ID de análise XY132002.

Iniciar um gerenciador de filas usando o IBM MQ Explorer ou usando o comando **amqmdain qmgr start** funciona corretamente porque esses comandos não iniciam diretamente o gerenciador de filas. Ao contrário, os comandos enviam um pedido para iniciar o gerenciador de filas em um processo separado em execução no espaço de nomes global.

Se os vários métodos de administração do IBM MQ não funcionarem ao usar serviços de terminal, tente configurar o direito de usuário `Criar objetos globais`.

Procedimento

1. Abra o painel Ferramentas Administrativas:

   Windows Server 2008 e Windows Server 2012
   Acesse esse painel usando **Painel de Controle** > **Sistema e Manutenção** > **Ferramentas Administrativas**.

   Windows 8.1
   Acesse esse painel utilizando **Ferramentas administrativas** > **Gerenciamento do computador**
2. Clique duas vezes em **Política de Segurança Local**.
3. Expanda **Políticas locais**.
4. Clique em **Designação de direitos do usuário**.
5. Inclua o novo usuário ou grupo na política `Criar objetos globais`.

Windows *O programa de saída do canal SSPI no Windows*

O IBM MQ for Windows fornece um programa de saída de segurança, que pode ser usado em canais de mensagens e canais de MQI. A saída é fornecida como código-fonte e de objeto, além de fornecer autenticação unilateral e de duas vias.

A saída de segurança usa o Security Support Provider Interface (SSPI), que fornece os recursos de segurança integrados das plataformas do Windows.

A saída de segurança fornece os seguintes serviços de identificação e de autenticação:

Autenticação de uma via
Este serviço usa o Windows NT suporte de autenticação do LAN Manager (NTLM). O NTLM permite que os servidores autentiquem seus clientes. Ele não permite que um cliente autentique um servidor, ou um servidor autentique outro servidor. O NTLM foi projetado para um ambiente de rede no qual supõe-se que os servidores sejam autênticos. NTLM é suportado em todas as plataformas do Windows que são suportadas pelo IBM WebSphere MQ 7.0.

Este serviço é usado geralmente em um canal de MQI para permitir que um gerenciador de filas do servidor autentique um aplicativo do IBM MQ MQI client. Um aplicativo cliente é identificado pelo ID de usuário associado ao processo em execução.

Para efetuar a autenticação, a saída de segurança na extremidade do cliente de um canal adquire um token de autenticação do NTLM e envia o token em uma mensagem de segurança para seu parceiro na outra extremidade do canal. A saída de segurança do parceiro transmite o token para o NTLM, que verifica se ele é autêntico. Se a saída de segurança do parceiro não for atendida quanto à autenticidade do token, ela instrui o MCA a fechar o canal.

Autenticação de duas vias ou mútua

Utiliza os serviços de autenticação do Kerberos. O protocolo Kerberos não supõe que os servidores em um ambiente de rede sejam autênticos. Servidores podem autenticar clientes e outros servidores, e clientes podem autenticar servidores. O Kerberos é suportado em todas as plataformas do Windows que são suportadas pelo IBM WebSphere MQ 7.0.

Este serviço pode ser utilizado em canais de mensagens e do MQI. Em um canal de mensagens, ele fornece autenticação mútua dos dois gerenciadores de filas. Em um canal de MQI, ele ativa o gerenciador de filas do servidor e o aplicativo do IBM MQ MQI client para autenticar um ao outro. Um gerenciador de filas é identificado por seu nome prefixado pela sequência `ibmMQSeries/`. Um aplicativo cliente é identificado pelo ID de usuário associado ao processo em execução.

Para efetuar a autenticação mútua, a saída de segurança iniciadora adquire um token de autenticação do servidor de segurança Kerberos e envia o token em uma mensagem de segurança para seu parceiro. A saída de segurança do parceiro transmite o token para o servidor de segurança do Kerberos, que verifica se é autêntico. O servidor de segurança do Kerberos gera um segundo token, que é enviado pelo parceiro em uma mensagem de segurança para a saída de segurança iniciadora. A saída de segurança iniciadora solicita então ao servidor Kerberos que verifique se o segundo token é autêntico. Durante esta troca, se alguma das saídas de segurança não for atendida quanto à autenticidade do token enviado pela outra, ela instrui o MCA a fechar o canal.

A saída de segurança é fornecida no formato de fonte e de objeto. Você pode utilizar o código fonte como um ponto de início para criar seus próprios programas de saída de canal ou utilizar o módulo de objeto conforme fornecido. O módulo de objeto tem dois pontos de entrada, uma para autenticação de uma via utilizando o suporte para autenticação do NTLM e o outro para autenticação de duas vias utilizando os serviços de autenticação do Kerberos.

Para obter mais informações sobre como o programa de saída do canal SSPI funciona e para obter instruções sobre como implementá-lo, consulte Usando a saída de segurança SSPI em sistemas Windows.

Windows *Aplicando arquivos de modelo de segurança no Windows*

Aplicar um modelo pode afetar as configurações de segurança aplicadas aos arquivos e diretórios do IBM MQ. Se você usa o modelo altamente seguro, aplique-o antes de instalar o IBM MQ.

O Windows suporta arquivos de modelo de segurança baseados em texto que podem ser usados para aplicar as configurações de segurança uniformes para um ou mais computadores com o snap-in MMC de configuração e análise de segurança. Em particular, o Windows fornece vários modelos que incluem um intervalo de configurações de segurança com o objetivo de fornecer níveis específicos de segurança. Esses modelos incluem Compatível, Seguro e Altamente Seguro.

Aplicar um desses modelos pode afetar as configurações de segurança aplicadas aos arquivos e diretórios do IBM MQ. Se quiser usar o modelo Altamente Seguro, configure sua máquina antes de instalar o IBM MQ.

Se aplicar o modelo altamente seguro a uma máquina na qual o IBM MQ já está instalada, todas as permissões definidas nos arquivos e diretórios do IBM MQ serão removidas. Com a remoção dessas permissões, você perde acesso ao grupo de *Administradores*, *mqm* e, quando aplicável, ao grupo *Todos*, a partir dos diretórios de erro.

Windows *Configurando autoridade adicional para aplicativos Windows que se conectam ao IBM MQ*

A conta sob a qual os processos do IBM MQ são executados talvez precise de autorização adicional para que o acesso SYNCHRONIZE aos processos do aplicativo possa ser concedido.

Sobre esta tarefa

Pode haver problemas se você tiver aplicativos do Windows, por exemplo, as páginas ASP, que se conectam ao IBM MQ que são configuradas para executar em um nível de segurança mais alto do que o normal.

O IBM MQ requer o acesso SYNCHRONIZE aos processos do aplicativo para coordenar determinadas ações. Quando um aplicativo do servidor tenta primeiro se conectar a um gerenciador de filas, o IBM MQ modifica o processo para conceder autoridade SYNCHRONIZE para os administradores do IBM MQ. No entanto, a conta sob a qual os processos IBM MQ são executados pode precisar de autorização adicional antes que o acesso solicitado possa ser concedido.

Para configurar autoridade adicional para o ID do usuário sob o qual os processos do IBM MQ estão em execução, conclua as seguintes etapas:

Procedimento

1. Inicie a ferramenta Política de segurança local, clique em **Configurações de segurança**->**Políticas locais**->**Designações de direito do usuário**, clique em **Depurar programas**.
2. Dê um clique duplo em **Programas de depuração** e, em seguida, inclua seu ID do usuário do IBM MQ na lista

   Se o sistema estiver em um domínio do Windows e a configuração de política efetiva ainda não estiver definida, mesmo que a configuração da política local esteja, o ID do usuário deve ser autorizado do mesmo modo no nível do domínio, usando a ferramenta Política de Segurança de Domínio.

IBM i Configurando a Segurança em IBM i

A segurança no IBM i é implementada usando o gerenciador de autoridade de objeto (OAM) do IBM MQ e a segurança no nível do objeto do IBM i.

Considerações de segurança que devem ser feitas ao determinar autoridade de acesso para objetos do IBM MQ.

Você precisa considerar os seguintes pontos ao configurar as autoridades para os usuários em sua empresa:

1. Conceda e revogue autoridades para os comandos do IBM MQ for IBM i usando os comandos IBM i GRTOBJAUT e RVKOBJAUT.

   Na biblioteca QMQM, certos objetos sem comando (*cmd) são configurados para ter autoridade ***PUBLIC** para ***USE**. Não altere as autoridades desses objetos, ou use uma lista de autorização para fornecer autoridade. Qualquer autoridade incorreta pode comprometer a funcionalidade do IBM MQ.
2. Durante a instalação do IBM MQ for IBM i, os seguintes perfis de usuário especial são criados:

   QMQM
   É usado principalmente para funções internas somente do produto. No entanto, ele pode ser usado para executar aplicativos confiáveis usando MQCNO_FASTPATH_BINDINGS. Consulte Conectando-se a um gerenciador de filas usando a chamada MQCONNX.

   QMQMADM
   É usado como um perfil de grupo para administradores do IBM MQ. O perfil do grupo fornece acesso a comandos CL e recursos do IBM MQ.

   Ao usar SBMJOB para enviar programas que chamam comandos do IBM MQ, USER não deve ser configurado explicitamente para QMQMADM. Em vez disso, configure USER para QMQM ou outro perfil do usuário que tenha QMQMADM especificado como um grupo.

3. Se você estiver enviando os comandos de canal para os gerenciadores de fila remotos, certifique-se de que o seu perfil de usuário é membro do grupo QMQMADM no sistema de destino. Para obter uma lista de comandos de canal PCF e MQSC, consulte IBM MQ for IBM i Comandos CL.
4. O conjunto de grupos associado a um usuário é armazenado em cache quando as autorizações do grupo são calculadas pelo OAM.

   As mudanças feitas nas associações do grupo de um usuário depois que o conjunto de grupos é armazenado em cache não são reconhecidas até que o gerenciador de filas seja reiniciado ou que RFRMQMAUT seja executado para atualizar a segurança.
5. Limite o número de usuários que possuem autoridade para trabalhar com comandos que são particularmente sigilosos. Esses comandos incluem:
   - Criar gerenciador da fila de mensagens (CRTMQM)
   - Excluir gerenciador da fila de mensagens (DLTMQM)
   - Iniciar gerenciador da fila de mensagens (STRMQM)
   - Finalizar gerenciador da fila de mensagens (ENDMQM)
   - Iniciar servidor de comandos (STRMQMCSVR)
   - Finalizar servidor de comandos (ENDMQMCSVR)
6. As definições de canal contêm uma especificação do programa de saída de segurança. A criação e a modificação do canal requerem considerações especiais. Os detalhes de saídas de segurança são fornecidos em "Visão Geral da Saída de Segurança" na página 116.
7. A saída de canal e os programas do monitor acionador podem ser substituídos. A segurança dessas substituições é de responsabilidade do programador.

IBM i Gerenciador de autoridade de objeto no IBM i

O gerenciador de autoridade de objeto (OAM) gerencia as autorizações dos usuários para manipular objetos do IBM MQ, incluindo filas e definições de processo. Ele também fornece uma interface de comandos pela qual é possível conceder ou revogar autoridade de acesso a um objeto para um grupo específico de usuários. A decisão de permitir acesso a um recurso é feita pelo OAM, e o gerenciador de filas segue essa decisão. Se o OAM não puder tomar uma decisão, o gerenciador de filas evitará o acesso a esse recurso.

Por meio do OAM, é possível controlar:

- O acesso a objetos do IBM MQ por meio do MQI. Quando um programa de aplicativo tenta acessar um objeto, o OAM verifica se o perfil do usuário que está fazendo a solicitação tem a autorização para a operação solicitada.

  Especificamente, isso significa que as filas, e as mensagens nas filas, podem ser protegidas contra acesso não autorizado.
- Permissão de usar comandos PCF e MQSC.

Diferentes grupos de usuários podem ter autoridade de acesso diferente para o mesmo objeto. Por exemplo, para uma fila específica, um grupo pode executar ambas as operações, put e get; outro grupo pode ter permissão apenas para navegar pela fila (MQGET com a opção de navegação). De forma semelhante, alguns grupos podem ter as autoridades get e put para uma fila, mas podem não ter permissão para alterar ou excluir a fila.

Comandos e operações de execução do IBM MQ for IBM i em objetos do IBM MQ for IBM i

IBM i Autoridades do IBM MQ no IBM i

Para acessar objetos do IBM MQ, é necessário ter a autoridade para emitir o comando e para acessar o objeto referenciado. Administradores têm acesso a todos os recursos do IBM MQ.

O acesso aos objetos do IBM MQ é controlado por autoridades para:

1. Emitir o comando do IBM MQ

2. Acessar os objetos do IBM MQ referenciados pelo comando

Todos os comandos de CL do IBM MQ for IBM i são fornecidos com um proprietário de QMQM, e a administração do perfil (QMQMADM) tem direitos *USE com o acesso *PUBLIC configurado como *EXCLUDE.

Nota: O programa QSRDUPER é usado pelo instalador do programa licenciado do IBM MQ for IBM i para duplicar objetos de Comando (*CMD) em QSYS. No IBM i V5R4 e posterior, o programa QSRDUPER foi mudado para que o comportamento padrão seja criar um comando proxy em vez de uma duplicação do comando original. Um comando proxy redireciona a execução do comando para outro comando e tem um atributo de PRX. Se existir um comando proxy com o mesmo nome do comando que está sendo copiado na biblioteca QSYS, as autoridades privadas para o comando proxy não serão concedidas ao comando na biblioteca do produto. Tentativas de solicitar ou de executar o comando proxy no QSYS verificam a autoridade do comando de destino na biblioteca do produto. Qualquer mudança na autoridade para objetos *CMD, portanto, precisa ser feita na biblioteca do produto (QMQM) e as do QSYS não precisam ser modificadas. Por exemplo:

```
GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
```

As mudanças na estrutura de autoridade de alguns dos comandos de CL do produto permitem o uso público destes comandos, se você tiver autoridade do OAM necessária para os objetos do IBM MQ para fazer essas mudanças.

Para ser um administrador do IBM MQ no IBM i, deve-se ser um membro do *grupo QMQMADM*. Esse grupo tem propriedades como as propriedades do grupo mqm em sistemas AIX, Linux, and Windows. Em particular, o grupo QMQMADM é criado ao se instalar o IBM MQ for IBM i, e os membros do grupo QMQMADM possuem acesso a todos os recursos do IBM MQ no sistema. Você também tem acesso a todos os recursos do IBM MQ se tiver a autoridade *ALLOBJ.

Os administradores podem usar comandos CL para administrar o IBM MQ. Um desses comandos de controle é GRTMQMAUT, que é utilizado para conceder autoridades a outros usuários. Outro comando, STRMQMMQSC, permite que um administrador emita comandos MQSC para um gerenciador de fila local.

Conceitos relacionados

“Autoridade para administrar o IBM MQ no IBM i” na página 95

IBM i *Autoridades de acesso para objetos do IBM MQ no IBM i*

As autoridades de acesso necessárias para executar os comandos de CL do IBM MQ.

O IBM MQ for IBM i categoriza os comandos de CL do produto em dois grupos:

Grupo 1

Os usuários devem estar no grupo de usuários QMQMADM ou ter a autoridade *ALLOBJ, para processar esses comandos. Usuários com uma dessas autoridades podem processar todos os comandos em todas as categorias sem precisar de autoridade extra.

Nota: Essas autoridades substituem qualquer autoridade OAM.

Estes comandos podem ser agrupados da seguinte forma:

- Comandos do Servidor de Comandos
  - ENDMQMCSVR, Finalizar o servidor de comandos do IBM MQ
  - STRMQMCSVR, Iniciar o servidor de comandos do IBM MQ
- Comando do Manipulador da Fila de Devoluções
  - STRMQMDLQ, Iniciar o manipulador da fila de devoluções do IBM MQ
- Comando Listener
  - ENDMQMLSR, Terminar listener IBM MQ
  - STRMQMLSR, Iniciar listener de não objeto
- Comandos de Recuperação de Mídia

  - RCDMQMIMG, Registrar imagem do objeto IBM MQ
  - RCRMQMOBJ, Recriar o objeto do IBM MQ
  - WRKMQMTRN, Trabalhar com as transações Q do IBM MQ
- Comandos do Gerenciador de Filas
  - CRTMQM, Criar Gerenciador da Fila de Mensagens
  - DLTMQM, Excluir Gerenciador da Fila de Mensagens
  - ENDMQM, Terminar Gerenciador da Fila de Mensagens
  - STRMQM, Iniciar Gerenciador da Fila de Mensagens
- Comandos de Segurança
  - GRTMQMAUT, Conceder autoridade de objeto ao IBM MQ
  - RVKMQMAUT, Revogar autoridade de objeto do IBM MQ
- Comando de Rastreio
  - TRCMQM, Rastrear tarefa do IBM MQ
- Comandos de Transação
  - RSVMQMTRN, Resolve IBM MQ Transaction
- Comandos do Monitor Acionador
  - STRMQMTRM, Iniciar Monitor Acionador
- Comandos de SC do IBM MQ
  - RUNMQSC, Executar comandos de SC do IBM MQ
  - STRMQMMQSC, Iniciar comandos de SC do IBM MQ

Grupo 2

O restante dos comandos, para os quais dois níveis de autoridade são necessários:

1. Autoridade do IBM i para executar o comando. Um administrador do IBM MQ configura isto usando o comando **GRTOBJAUT** para substituir a restrição *PUBLIC(de *EXCLUDE) em um usuário ou grupo de usuários.

   Por exemplo:

   ```
   GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
   ```

2. Autoridade do IBM MQ para manipular os objetos do IBM MQ associados ao comando, ou comandos, dada a autoridade correta do IBM i na Etapa 1.

   Esta autoridade é controlada pelo usuário que tem a autoridade OAM apropriada para a ação necessária, definida por um administrador do IBM MQ usando o comando **GRTMQMAUT**

   Por exemplo:

   ```
   GRTMQMAUT *connect authority to the queue manager + *admchg authority to
           the queue
   ```

Os comandos podem ser agrupados da seguinte forma:

- Comandos do Canal
  - CHGMQMCHL, Alterar Canal IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas e a autoridade *admchg para o canal.
  - CPYMQMCHL, Copiar Canal IBM MQ

Isso requer a autoridade *connect e *admcrt para o gerenciador de filas, a autoridade *admdsp para o tipo de canal padrão a ser copiado e a autoridade *admcrt para a classe de objeto do canal.

Por exemplo, copiar um canal emissor precisa da autoridade *admdsp para o canal SYSTEM.DEF.SENDER

– CRTMQMCHL, Criar Canal IBM MQ

Isso requer a autoridade *connect e *admcrt para o gerenciador de filas, a autoridade *admdsp para o tipo de canal padrão a ser criado e a autoridade *admcrt para a classe de objeto do canal.

Por exemplo, criar um canal emissor precisa da autoridade *admdsp para o canal SYSTEM.DEF.SENDER.

– DLTMQMCHL, Excluir Canal IBM MQ

Isso requer a autoridade *connect para o gerenciador de filas, e a autoridade *admdlt para o canal.

– RSVMQMCHL, Resolver IBM MQ Canal

Isso requer a autoridade *connect para o gerenciador de filas, e a autoridade *ctrlx para o canal.

- Comandos de Exibição

Para processar os comandos DSP, você deve conceder ao usuário as autoridades `*connect` e `*admdsp` para o gerenciador de filas, juntamente com qualquer opção específica listada:

– DSPMQM, Exibir Gerenciador da Fila de Mensagens

– DSPMQMAUT, Display IBM MQ Object Authority

– DSPMQMAUTI, Display IBM MQ Authentication Information - `*admdsp` para o objeto de informações sobre autenticação

– DSPMQMCHL, Display IBM MQ Channel - `*admdsp` para o canal

– DSPMQMCSVR, Exibir Servidor de Comandos IBM MQ

– DSPMQMNL, Display IBM MQ Namelist - `*admdsp` para a lista de nomes

– DSPMQMOBJN, Display IBM MQ Object Names

– DSPMQMPRC, Display IBM MQ Process - `*admdsp` para o processo

– DSPMQMQ, Display IBM MQ Queue - `*admdsp` para a fila

– DSPMQMTOP, Display IBM MQ Topic - `*admdsp` para o tópico

- Trabalhar com comandos

Para processar os comandos WRK e exibir o painel de opções, você deve conceder ao usuário as autoridades `*connect` e `*admdsp` para o gerenciador de filas, juntamente com qualquer opção específica listada:

– WRKMQM, Trabalhar com Gerenciadores da Fila de Mensagens

– WRKMQMAUT, Trabalhar com a Autoridade do Objeto IBM MQ

– WRKMQMAUTD, Trabalhar com Dados de Autoridade do Objeto IBM MQ

– WRKMQMAUTI, Trabalhar com IBM MQ Informações sobre Autenticação

- `*admchg` para o comando Change IBM MQ Authentication Information Object.
- `*admcrt` para o comando Create and Copy IBM MQ Authentication Information Object.
- `*admdlt` para o comando Delete IBM MQ Authentication Information Object.
- `*admdsp` para o comando Display IBM MQ Authentication Information Object.

– WRKMQMCHL, Trabalhar com o canal do IBM MQ

Isso requer as seguintes autoridades:

- `*admchg` para o comando Change IBM MQ Channel.

- *admclr para o comando Clear IBM MQ Channel.
- *admcrt para o comando Create and Copy IBM MQ Channel.
- *admdlt para o comando Delete IBM MQ Channel.
- *admdsp para o comando Display IBM MQ Channel.
- *ctrl para o comando Start IBM MQ Channel.
- *ctrl para o comando End IBM MQ Channel.
- *ctrl para o comando Ping IBM MQ Channel.
- *ctrlx para o comando Reset IBM MQ Channel.
- *ctrlx para o comando Resolve IBM MQ Channel.

– WRKMQMCHST, Trabalhar com o Status do Canal IBM MQ

  Isso requer a autoridade *admdsp para o canal.
– WRKMQMCL, Trabalhar com Clusters IBM MQ
– WRKMQMCLQ, Trabalhar com IBM MQ Filas de Cluster
– WRKMQMCLQM, Trabalhar com o Gerenciador de Filas do Cluster IBM MQ
– WRKMQMLSR, Trabalhar com Listener IBM MQ
– WRKMQMMSG, Trabalhar com Mensagens IBM MQ

  Isso requer a autoridade *browse para a fila
– WRKMQMNL, Trabalhar com listas de nomes do IBM MQ

  Isso requer as seguintes autoridades:
  - *admchg para o comando Change IBM MQ Namelist.
  - *admcrt para o comando Create and Copy IBM MQ Namelist.
  - *admdlt para o comando Delete IBM MQ Namelist.
  - *admdsp para o comando Display IBM MQ Namelist.
– WRKMQMPRC, Trabalhar com processos do IBM MQ

  Isso requer as seguintes autoridades:
  - *admchg para o comando Change IBM MQ Process.
  - *admcrt para o comando Create and Copy IBM MQ Process.
  - *admdlt para o comando Delete IBM MQ Process.
  - *admdsp para o comando Display IBM MQ Process.
– WRKMQMQ, Trabalhar com filas do IBM MQ

  Isso requer as seguintes autoridades:
  - *admchg para o comando Change IBM MQ Queue.
  - *admclr para o comando Clear IBM MQ Queue.
  - *admcrt para o comando Create and Copy IBM MQ Queue.
  - *admdlt para o comando Delete IBM MQ Queue.
  - *admdsp para o comando Display IBM MQ Queue.
– WRKMQMQSTS, Trabalhar com IBM MQ Status da fila
– WRKMQMTOP, Trabalhar com IBM MQ Tópicos

  Isso requer as seguintes autoridades
  - *admchg para o comando Change IBM MQ Topic.
  - *admcrt para o comando Create and Copy IBM MQ Topic.
  - *admdlt para o comando Delete IBM MQ Topic.

- *admdsp para o comando Display IBM MQ Topic.

– WRKMQMSUB, Trabalhar com assinaturas do IBM MQ

- Outros comandos de Canal

  Para processar os comandos de canal, você deve conceder ao usuário as autoridades específicas listadas:

  – ENDMQMCHL, Fim do Canal IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas e a autoridade *allmqi para a fila de transmissão associada ao canal.

  – ENDMQMLSR, Encerrar Listener IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas e a autoridade *ctrl para o objeto de listener nomeado.

  – PNGMQMCHL, Ping IBM MQ Canal

    Isto requer a autoridade *connect e *inq para o gerenciador de filas, e a autoridade *ctrl para o objeto do canal.

  – RSTMQMCHL, Reconfigurar Canal IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas.

  – STRMQMCHL, Iniciar Canal IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas e a autoridade *ctrl para o objeto de canal.

  – STRMQMCHLI, Iniciar Inicializador de Canais IBM MQ

    Isso requer as autoridades *connect e *inq para o gerenciador de filas e a autoridade *allmqi para a fila de inicialização associada à fila de transmissão do canal.

  – STRMQMLSR, Iniciar Listener IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas e a autoridade *ctrl para o objeto de listener nomeado.

- Outros comandos:

  Para processar os seguintes comandos, você deve conceder ao usuário as autoridades específicas listadas:

  – CCTMQM, Conectar-se ao Gerenciador da Fila de Mensagens

    Isso não requer autoridade de objeto do IBM MQ.

  – CHGMQM, Alterar Gerenciador da Fila de Mensagens

    Isso requer as autoridades *connect e *admchg para o gerenciador de filas.

  – CHGMQMAUTI, Alterar IBM MQ Informações sobre Autenticação

    Isso requer a autoridade *connect para o gerenciador de filas, e a autoridade *admchg e *admdsp para o objeto de informações sobre autenticação.

  – CHGMQMNL, Alterar Lista de Nomes IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas e a autoridade *admchg para a lista de nomes.

  – CHGMQMPRC, Alterar Processo IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas e a autoridade *admchg para o processo.

  – CHGMQMQ, Alterar Fila IBM MQ

    Isso requer a autoridade *connect para o gerenciador de filas e a autoridade *admchg para a fila.

– CLRMQMQ, Limpar Fila IBM MQ

  Isso requer a autoridade `*connect` para o gerenciador de filas e a autoridade `*admclr` para a fila.

– CPYMQMAUTI, Copiar IBM MQ Informações sobre Autenticação

  Isso requer a autoridade `*connect` para o gerenciador de filas, a autoridade `*admdsp` para o objeto de informações sobre autenticação e a autoridade `*admcrt` para a classe do objeto de informações sobre autenticação.

– CPYMQMNL, Lista de Nomes de Cópia IBM MQ

  Isso requer as autoridades `*connect` e `*admcrt` para o gerenciador de filas.

– CPYMQMPRC, Copiar IBM MQ Processo

  Isso requer as autoridades `*connect` e `*admcrt` para o gerenciador de filas.

– CPYMQMQ, Fila de Cópia IBM MQ

  Isso requer as autoridades `*connect` e `*admcrt` para o gerenciador de filas.

– CRTMQMAUTI, Criar IBM MQ Informações sobre Autenticação

  Isso requer a autoridade `*connect` para o gerenciador de filas, a autoridade `*admdsp` para o objeto de informações sobre autenticação e a autoridade `*admcrt` para a classe do objeto de informações sobre autenticação.

– CRTMQMNL, Criar Lista de Nomes IBM MQ

  Isso requer as autoridades `*connect` e `*admcrt` para o gerenciador de filas e a autoridade `*admdsp` para a lista de nomes padrão.

– CRTMQMPRC, Criar Processo IBM MQ

  Isso requer as autoridades `*connect` e `*admcrt` para o gerenciador de filas e a autoridade `*admdsp` para o processo padrão.

– CRTMQMQ, Criar Fila IBM MQ

  Isso requer as autoridades `*connect` e `*admcrt` para o gerenciador de filas e a autoridade `*admdsp` para a fila padrão.

– CVTMQMDTA, Converter Comando de Tipo de Dados IBM MQ

  Isso não requer autoridade de objeto do IBM MQ.

– DLTMQMAUTI, Excluir Informações sobre Autenticação IBM MQ

  Isso requer a autoridade `*connect` para o gerenciador de filas, e a autoridade `*ctrlx` para o objeto de informações sobre autenticação.

– DLTMQMNL, Excluir Namelist IBM MQ

  Isso requer a autoridade `*connect` para o gerenciador de filas e a autoridade `*admdlt` para a lista de nomes.

– DLTMQMPRC, Excluir IBM MQ Processo

  Isso requer a autoridade `*connect` para o gerenciador de filas e a autoridade `*admdlt` para o processo.

– DLTMQMQ, Excluir Fila IBM MQ

  Isso requer a autoridade `*connect` para o gerenciador de filas e a autoridade `*admdlt` para a fila.

– DSCMQM, Desconectar-se do Gerenciador da Fila de Mensagens

  Isso não requer autoridade de objeto do IBM MQ.

– RFRMQMAUT, Atualizar Segurança

  Isso requer a autoridade `*connect` para o gerenciador de filas.

– RFRMQMCL, Atualizar Cluster

  Isso requer a autoridade `*connect` para o gerenciador de filas.
– RSMMQMCLQM, Continuar o Gerenciador de Filas do Cluster

  Isso requer a autoridade `*connect` para o gerenciador de filas.
– RSTMQMCL, Reconfigurar Cluster

  Isso requer a autoridade `*connect` para o gerenciador de filas.
– SPDMQMCLQM, Suspender Gerenciador de Filas do Cluster

  Isso requer a autoridade `*connect` para o gerenciador de filas.

IBM i *Autorizações de acesso no IBM i*

Use estas informações para entender os comandos de autorização de acesso.

As autorizações definidas pela palavra-chave AUT nos comandos GRTMQMAUT e RVKMQMAUT podem ser categorizadas da seguinte forma:

- Autorizações relacionadas a chamadas MQI
- Comandos de administração relacionados à autorização
- Autorizações de contexto
- Autorizações gerais, isto é, para chamadas MQI, para comandos, ou ambos

As tabelas a seguir listam as diferentes autoridades, usando o parâmetro AUT para chamadas MQI, chamadas de Contexto, comandos MQSC e PCF e operações genéricas.

Tabela 15. Autorizações para Chamadas MQI

AUT	Descrição
`*ALTUSR`	Permitir que a autoridade de um outro usuário seja usada para chamadas de MQOPEN e MQPUT1.
`*BROWSE`	Recuperar uma mensagem de uma fila, emitindo uma chamada MQGET com a opção BROWSE.
`*CONNECT`	Conecte o aplicativo ao gerenciador de filas especificado, emitindo uma chamada MQCONN.
`*GET`	Recuperar uma mensagem de uma fila, emitindo uma chamada MQGET.
`*INQ`	Fazer uma consulta em uma fila específica, emitindo uma chamada MQINQ.
`*PUB`	Abrir um tópico para publicar uma mensagem, usando uma chamada MQPUT.
`*PUT`	Colocar uma mensagem em uma fila específica, emitindo uma chamada MQPUT.
`*RESUME`	Continuar uma assinatura, usando uma chamada MQSUB.
`*SET`	Configurar atributos em uma fila a partir de MQI, emitindo uma chamada MQSET. Se você abrir uma fila para várias opções, deverá ter autorização para cada uma delas.
`*SUB`	Criar, Alterar ou Continuar uma assinatura para um tópico, usando uma chamada MQSUB.

Tabela 16. Autorizações para Chamadas de Contexto

AUT	Descrição
`*PASSALL`	Passar todo o contexto na fila especificada. Todos os campos de contexto são copiados da solicitação original.

Tabela 16. Autorizações para Chamadas de Contexto (continuação)	
AUT	**Descrição**
*PASSID	Passar contexto de identidade na fila especificada. O contexto de identidade é igual àquele da solicitação.
*SETALL	Configurar todo o contexto na fila especificada. Isso é usado por utilitários especiais do sistema.
*SETID	Configurar contexto de identidade na fila especificada. Isso é usado por utilitários especiais do sistema.

Tabela 17. Autorizações para Chamadas MQSC e PCF	
AUT	**Descrição**
*ADMCHG	Alterar os atributos do objeto especificado.
*ADMCLR	Limpar o objeto especificado (apenas o comando PCF Limpar objeto).
*ADMCRT	Criar objetos do tipo especificado.
*ADMDLT	Excluir o objeto especificado.
*ADMDSP	Exibir os atributos do objeto especificado.

Tabela 18. Autorizações para Operações Genéricas	
AUT	**Descrição**
*ALL	Usar todas as operações aplicáveis ao objeto. A autoridade `all` é equivalente à união das autoridades `alladm`, `allmqi` e `system` apropriadas ao tipo de objeto.
*ALLADM	Executar todas as operações de administração aplicáveis ao objeto.
*ALLMQI	Usar todas as chamadas MQI aplicáveis ao objeto.
*CTRL	Controlar a inicialização e o encerramento de canais, listeners e serviços.
*CTRLX	Reconfigurar o número de sequência e resolver canais indeterminados.

IBM i *Usando os comandos de autorização de acesso no IBM i*

Use estas informações para aprender sobre os comandos de autorização de acesso e use os exemplos de comandos.

Usando o Comando GRTMQMAUT

Se você tiver a autorização necessária, poderá usar o comando GRTMQMAUT para conceder autorização de um perfil de usuário ou grupo de usuários para acessar um determinado objeto. Os seguintes exemplos ilustram como o comando GRTMQMAUT é usado:

1.
```
GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
          AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
```

   Neste exemplo:

   - RED.LOCAL.QUEUE é o nome do objeto.
   - *LCLQ (fila local) é o tipo de objeto.
   - GROUPA é o nome de um perfil de usuário no sistema para o qual as autorizações devem ser mudadas. Esse perfil pode ser usado como um perfil do grupo para outros usuários.
   - *BROWSE e *PUT são as autorizações que estão sendo concedidas à fila especificada.

*BROWSE inclui autorização para navegar pelas mensagens na fila (para emitir MQGET com a opção de navegação).

*PUT inclui autorização para colocar (MQPUT) mensagens na fila.

- `saturn.queue.manager` é o nome do gerenciador de filas.

2. O comando a seguir concede aos usuários JACK e JILL todas as autorizações aplicáveis, a todas as definições de processos, para o gerenciador de filas padrão.

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
```

3. O comando a seguir concede ao usuário GEORGE autoridade para colocar uma mensagem na fila ORDERS, no gerenciador de filas TRENT.

```
GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
```

Usando o Comando RVKMQMAUT

Se você tiver a autorização necessária, poderá usar o comando RVKMQMAUT para remover a autorização concedida anteriormente de um perfil de usuário ou grupo de usuários para acessar um determinado objeto. Os seguintes exemplos ilustram como o comando RVKMQMAUT é usado:

1.
```
RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
AUT(*PUT) MQMNAME('saturn.queue.manager')
```

   A autoridade para colocar mensagens na fila especificada, que foi concedida no exemplo anterior, é removida de GROUPA.

2.
```
RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
MQMNAME(PAYROLLQM)
```

   A autoridade para obter mensagens de qualquer fila com um nome que começa com os caracteres PAY, pertencente ao gerenciador de filas PAYROLLQM, é removida de todos os usuários do sistema, a menos que eles, ou um grupo ao qual eles pertencem, tenham sido autorizados separadamente.

Usando o Comando DSPMQMAUT

O comando (DSPMQMAUT) da autoridade de MQM mostra, para o objeto e usuário especificados, a lista de autorizações que o usuário tem para o objeto. O seguinte exemplo ilustra como o comando é usado:

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

Usando o Comando RFRMQMAUT

O comando de segurança do MQM de atualização (RFRMQMAUT) permite atualizar as informações do grupo de autorização do OAM imediatamente, refletindo as mudanças feitas no nível do sistema operacional, sem precisar parar e reiniciar o gerenciador de filas. O seguinte exemplo ilustra como o comando é usado:

```
RFRMQMAUT MQMNAME(ADMINQM)
```

IBM i Tabelas de especificação de autorização no IBM i

Use estas informações para determinar qual autorização é necessária para usar chamadas API específicas, e opções específicas dessas chamadas, em objetos de fila, objetos de processo e objetos de gerenciador de filas.

As tabelas de especificação de autorização, iniciadas na Tabela 19 na página 175, definem precisamente como as autorizações funcionam e as restrições que se aplicam. As tabelas aplicam-se a estas situações:

- Aplicativos que emitem chamadas MQI
- Programas de administração que emitem comandos MQSC como PCFs Escape
- Programas de administração que emitem comando de PCF

Nesta seção, as informações são apresentadas como um conjunto de tabelas que especificam os seguintes dados:

Ação a ser executada
Opção de MQI, comando MQSC ou comando PCF.

Objeto de controle de acesso
Fila, definição de processo, gerenciador de filas, lista de nomes, canal, canal de conexão do cliente, listener, serviço ou objeto de informações sobre autenticação.

Authorization required
Expressa como uma constante MQZAO_.

Nas tabelas, as constantes prefixadas com MQZAO_ correspondem às palavras-chave na lista de autorização para os comandos **GRTMQMAUT** e **RVKMQMAUT** da entidade específica. Por exemplo, MQZAO_BROWSE corresponde à palavra-chave *BROWSE; de forma semelhante, a palavra-chave MQZAO_SET_ALL_CONTEXT corresponde à palavra-chave *SETALL e assim por diante. Essas constantes são definidas no arquivo de cabeçalho cmqzc.h, fornecido com o produto.

autorizações MQI

Um aplicativo terá permissão para emitir chamadas MQI e opções específicas somente se o identificador de usuário sob o qual estiver sendo executado (ou cujas autorizações puder assumir) tiver recebido a autorização relevante.

Quatro chamadas MQI requerem verificações de autorização: MQCONN, MQOPEN, MQPUT1 e MQCLOSE.

Para MQOPEN e MQPUT1, a verificação de autoridade é feita no nome do objeto que está sendo aberto e não no nome ou nomes resultantes após a resolução de um nome. Por exemplo, um aplicativo pode receber autoridade para abrir uma fila de alias sem ter autoridade para abrir a fila de base para a qual o alias é resolvido. A regra é que a verificação seja realizada na primeira definição encontrada durante o processo de resolução do nome que não seja um alias do gerenciador de filas, a menos que a definição de alias do gerenciador de filas seja aberta diretamente; ou seja, seu nome apareça no campo *ObjectName* do descritor de objeto. A autoridade é sempre necessária para o objeto específico que está sendo aberto; em alguns casos, a autoridade independente de fila adicional, obtida por meio de uma autorização para o objeto do gerenciador de filas, é necessária.

Tabela 19 na página 175, Tabela 20 na página 175, Tabela 21 na página 176 e Tabela 22 na página 176 resumem as autorizações necessárias para cada chamada.

Nota: Essas tabelas não mencionam listas de nomes, canais, canais de conexão do cliente, listeners, serviços ou objetos de informações sobre autenticação. Isso é porque nenhuma das autorizações se aplica a esses objetos, exceto MQOO_INQUIRE, para o qual as mesmas autorizações se aplicam como para os outros objetos.

Tabela 19. Autorização de segurança necessária para chamadas MQCONN			
Autorização necessária para:	**Objeto da fila (“1” na página 176)**	**Objeto de processo**	**Objeto do gerenciador de filas**
opção MQCONN	Não-aplicável	Não-aplicável	MQZAO_CONNECT

Tabela 20. Autorização de segurança necessária para chamadas MQOPEN			
Autorização necessária para:	**Objeto da fila (“1” na página 176)**	**Objeto de processo**	**Objeto do gerenciador de filas**
MQOO_INQUIRE	MQZAO_INQUIRE **(“2” na página 176)**	MQZAO_INQUIRE **(“2” na página 176)**	MQZAO_INQUIRE **(“2” na página 176)**
MQOO_BROWSE	MQZAO_BROWSE	Não-aplicável	Sem verificação
MQOO_INPUT_*	MQZAO_INPUT	Não-aplicável	Sem verificação
MQOO_SAVE_ ALL_CONTEXT **(“3” na página 176)**	MQZAO_INPUT	Não-aplicável	Não-aplicável
MQOO_OUTPUT (fila Normal) **(“4” na página 176)**	MQZAO_OUTPUT	Não-aplicável	Não-aplicável
MQOO_PASS_ IDENTITY_CONTEXT **(“5” na página 176)**	MQZAO_PASS_ IDENTITY_CONTEXT	Não-aplicável	Sem verificação
MQOO_PASS_ALL_ CONTEXT **(“5” na página 176, “6” na página 176)**	MQZAO_PASS _ALL_CONTEXT	Não-aplicável	Sem verificação
MQOO_SET_ IDENTITY_CONTEXT **(“5” na página 176, “6” na página 176)**	MQZAO_SET_ IDENTITY_CONTEXT	Não-aplicável	MQZAO_SET_ IDENTITY_CONTEXT **(“7” na página 176)**
MQOO_SET_ ALL_CONTEXT **(“5” na página 176, “8” na página 177)**	MQZAO_SET_ ALL_CONTEXT	Não-aplicável	MQZAO_SET_ ALL_CONTEXT **(“7” na página 176)**
MQOO_OUTPUT (Fila de transmissão) **(“9” na página 177)**	MQZAO_SET_ ALL_CONTEXT	Não-aplicável	MQZAO_SET_ ALL_CONTEXT **(“7” na página 176)**
MQOO_SET	MQZAO_SET	Não-aplicável	Sem verificação
MQOO_ALTERNATE_ USER_AUTHORITY	**(“10” na página 177)**	**(“10” na página 177)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(“10” na página 177, “11” na página 177)**

Tabela 21. Autorização de segurança necessária para chamadas MQPUT1			
Autorização necessária para:	**Objeto da fila (“1” na página 176)**	**Objeto de processo**	**Objeto do gerenciador de filas**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(“12” na página 177)**	Não-aplicável	Sem verificação
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(“12” na página 177)**	Não-aplicável	Sem verificação
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(“12” na página 177)**	Não-aplicável	MQZAO_SET_ IDENTITY_CONTEXT **(“7” na página 176)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(“12” na página 177)**	Não-aplicável	MQZAO_SET_ ALL_CONTEXT **(“7” na página 176)**
(Fila de transmissão) **(“9” na página 177)**	MQZAO_SET_ ALL_CONTEXT	Não-aplicável	MQZAO_SET_ ALL_CONTEXT **(“7” na página 176)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(“13” na página 177)**	Não-aplicável	MQZAO_ALTERNATE_ USER_AUTHORITY **(“11” na página 177)**

Tabela 22. Autorização de segurança necessária para chamadas MQCLOSE			
Autorização necessária para:	**Objeto da fila (“1” na página 176)**	**Objeto de processo**	**Objeto do gerenciador de filas**
MQCO_DELETE	MQZAO_DELETE **(“14” na página 177)**	Não-aplicável	Não-aplicável
MQCO_DELETE _PURGE	MQZAO_DELETE **(“14” na página 177)**	Não-aplicável	Não-aplicável

Notas para as tabelas:

1. Se uma fila modelo estiver sendo aberta:
   - A autoridade MQZAO_DISPLAY será necessária para a fila modelo, além da autoridade para abrir a fila modelo para o tipo de acesso para o qual você está abrindo.
   - A autoridade MQZAO_CREATE não é necessária para criar o fila dinâmica.
   - O identificador de usuários usado para abrir a fila modelo recebe automaticamente todas as autoridades específicas da fila (equivalente a MQZAO_ALL) para a fila dinâmica criada.
2. O objeto de fila, de processo, de lista de nomes ou de gerenciador de filas é verificado, dependendo do tipo de objeto que está sendo aberto.
3. MQOO_INPUT_* também deve ser especificado. Essa opção é válida para uma fila local, modelo ou de alias.
4. Essa verificação é executada para todos os casos de saída, exceto para o caso especificado na nota “9” na página 177.
5. MQOO_OUTPUT também deve ser especificado.
6. MQOO_PASS_IDENTITY_CONTEXT também é sugerido por essa opção.
7. Essa autoridade é necessária para o objeto de gerenciador de filas e a fila específica.

8. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT e MQOO_SET_IDENTITY_CONTEXT também são sugeridos por essa opção.
9. Essa verificação é executada para uma fila local ou modelo que tem um atributo de fila *Usage* de MQUS_TRANSMISSION e está sendo aberta diretamente para saída. Ela não será aplicada se uma fila remota estiver sendo aberta (especificando-se os nomes do gerenciador de filas remotas e da fila remota ou especificando-se o nome de uma definição local da fila remota).
10. Pelo menos um de MQOO_INQUIRE (para qualquer tipo de objeto) ou (para filas) MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT ou MQOO_SET também deve ser especificado. A verificação executada é como para as outras opções especificadas, usando-se o identificador de usuário alternativo fornecido para a autoridade de objeto com nome específico, e a autoridade de aplicativo atual para a verificação MQZAO_ALTERNATE_USER_IDENTIFIER.
11. Essa autorização permite que qualquer *AlternateUserId* seja especificado.
12. Uma verificação MQZAO_OUTPUT também será executada se a fila não tiver um atributo de fila *Usage* de MQUS_TRANSMISSION.
13. A verificação executada é como para as outras opções especificadas, usando-se o identificador de usuário alternativo fornecido para a autoridade de fila nomeada, e a autoridade de aplicativo atual para a verificação MQZAO_ALTERNATE_USER_IDENTIFIER.
14. A verificação será realizada somente se ambas as instruções a seguir forem verdadeiras:
    - Uma fila dinâmica permanente está sendo fechada e excluída.
    - A fila não foi criada pela MQOPEN que retornou a manipulação de objetos que está sendo usada.

    Caso contrário, não haverá verificação.

Notas gerais:

1. A autorização especial MQZAO_ALL_MQI inclui todas as seguintes autorizações que são relevantes ao tipo de objeto:
   - MQZAO_CONNECT
   - MQZAO_INQUIRE
   - MQZAO_SET
   - MQZAO_BROWSE
   - MQZAO_INPUT
   - MQZAO_OUTPUT
   - MQZAO_PASS_IDENTITY_CONTEXT
   - MQZAO_PASS_ALL_CONTEXT
   - MQZAO_SET_IDENTITY_CONTEXT
   - MQZAO_SET_ALL_CONTEXT
   - MQZAO_ALTERNATE_USER_AUTHORITY
2. MQZAO_DELETE (consulte a nota “14” na página 177) e MQZAO_DISPLAY são classificados como autorizações de administração. Portanto, elas não são incluídas em MQZAO_ALL_MQI.
3. *Nenhuma verificação* significa que nenhuma verificação de autorização é executada.
4. *Não aplicável* significa que a verificação de autorização não é relevante para essa operação. Por exemplo, não é possível emitir uma chamada MQPUT para um objeto de processo.

IBM i *Autorizações para comandos MQSC em PCFs de escape no IBM i*

Essas autorizações permitem que um usuário emita comandos de administração como uma mensagem PCF Escape. Esses métodos permitem que um programa envie um comando de administração como uma mensagem para um gerenciador de filas, para execução em nome desse usuário.

Esta seção resume as autorizações necessárias para cada comando MQSC contido no PCF Escape.

Não aplicável significa que a verificação de autorização não é relevante para essa operação.

O ID de usuário com o qual o programa que envia o comando está sendo executado também deve ter as seguintes autoridades:

- Autoridade MQZAO_CONNECT para o gerenciador de filas
- Autoridade DISPLAY no gerenciador de filas para executar comando de PCF
- Autoridade para emitir os comandos MQSC dentro do texto do comando PCF Escape

ALTER *object*

Object	**Authorization required**
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	MQZAO_CHANGE
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Listener	MQZAO_CHANGE
Serviço	MQZAO_CHANGE

CLEAR *object*

Object	**Authorization required**
Fila	MQZAO_CLEAR
Tópico	MQZAO_CLEAR
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	Não-aplicável
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

DEFINE *object* NOREPLACE (“1” na página 182)

Object	**Authorization required**
Fila	MQZAO_CREATE **(“2” na página 182)**
Tópico	MQZAO_CREATE **(“2” na página 182)**
Processo	MQZAO_CREATE **(“2” na página 182)**
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CREATE **(“2” na página 182)**

Object	Authorization required
Informações sobre Autenticação	MQZAO_CREATE **("2" na página 182)**
Canal	MQZAO_CREATE **("2" na página 182)**
Canal de conexão do cliente	MQZAO_CREATE **("2" na página 182)**
Listener	MQZAO_CREATE **("2" na página 182)**
Serviço	MQZAO_CREATE **("2" na página 182)**

DEFINE *object* REPLACE ("1" na página 182, "3" na página 182)

Object	Authorization required
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Listener	MQZAO_CHANGE
Serviço	MQZAO_CHANGE

EXCLUIR *object*

Object	Authorization required
Fila	MQZAO_DELETE
Tópico	MQZAO_DELETE
Processo	MQZAO_DELETE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_DELETE
Informações sobre Autenticação	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de conexão do cliente	MQZAO_DELETE
Listener	MQZAO_DELETE
Serviço	MQZAO_DELETE

DISPLAY *object*

Object	Authorization required
Fila	MQZAO_DISPLAY
Tópico	MQZAO_DISPLAY
Processo	MQZAO_DISPLAY

Object	Authorization required
Gerenciador de filas	MQZAO_DISPLAY
Lista de Nomes	MQZAO_DISPLAY
Informações sobre Autenticação	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de conexão do cliente	MQZAO_DISPLAY
Listener	
Serviço	

PING CHANNEL

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

RESET CHANNEL

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL_EXTENDED
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

RESOLVE CHANNEL

Object	Authorization required
Fila	Não-aplicável

Object	Authorization required
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL_EXTENDED
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

START *object*

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Listener	MQZAO_CONTROL
Serviço	MQZAO_CONTROL

PARAR *object*

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Listener	MQZAO_CONTROL
Serviço	MQZAO_CONTROL

Nota:

1. Para comandos DEFINE, a autoridade MQZAO_DISPLAY também será necessária para o objeto LIKE, se houver um especificado, ou no objeto SYSTEM.DEFAULT.xxx apropriado, se LIKE for omitido.
2. A autoridade MQZAO_CREATE não é específica a um determinado objeto ou tipo de objeto. A autoridade Criar é concedida para todos os objetos de um gerenciador de filas especificado, especificando-se um tipo de objeto de QMGR no comando GRTMQMAUT.
3. Essa opção se aplicará se o objeto a ser substituído já existir. Caso contrário, a verificação será como para DEFINE *object* NOREPLACE.

IBM i *Autorizações para comandos de PCF no IBM i*

Essas autorizações permitem que um usuário emita comandos de administração como comando de PCF. Esses métodos permitem que um programa envie um comando de administração como uma mensagem para um gerenciador de filas, para execução em nome desse usuário.

Esta seção resume as autorizações necessárias para cada comando PCF.

Nenhuma verificação significa que nenhuma verificação de autorização é executada; *Não aplicável* significa que a verificação de autorização não é relevante para essa operação.

O ID de usuário com o qual o programa que envia o comando está sendo executado também deve ter as seguintes autoridades:

- Autoridade MQZAO_CONNECT para o gerenciador de filas
- Autoridade DISPLAY no gerenciador de filas para executar comando de PCF

A autorização especial MQZAO_ALL_ADMIN inclui as autorizações a seguir:

- MQZAO_CHANGE
- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY
- MQZAO_CONTROL
- MQZAO_CONTROL_EXTENDED

A autoridade MQZAO_CREATE não é específica a um determinado objeto ou tipo de objeto

Mudar *object*

Object	**Authorization required**
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	MQZAO_CHANGE
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Listener	MQZAO_CHANGE
Serviço	MQZAO_CHANGE

Clear *object*

Object	Authorization required
Fila	MQZAO_CLEAR
Tópico	MQZAO_CLEAR
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	Não-aplicável
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

Copiar *object* (sem substituição) (“1” na página 187)

Object	Authorization required
Fila	MQZAO_CREATE **(“2” na página 188)**
Tópico	MQZAO_CREATE **(“2” na página 188)**
Processo	MQZAO_CREATE **(“2” na página 188)**
Gerenciador de filas	Não-aplicável
NamelistMQZAO_CREATE	MQZAO_CREATE **(“2” na página 188)**
Informações sobre Autenticação	MQZAO_CREATE **(“2” na página 188)**
Canal	MQZAO_CREATE **(“2” na página 188)**
Canal de conexão do cliente	MQZAO_CREATE **(“2” na página 188)**
Listener	MQZAO_CREATE **(“2” na página 188)**
Serviço	MQZAO_CREATE **(“2” na página 188)**

Copiar *object* (com substituição) (“1” na página 187, “4” na página 188)

Object	Authorization required
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Listener	MQZAO_CHANGE

Object	Authorization required
Serviço	MQZAO_CHANGE

Criar *object* (sem substituição) (“3” na página 188)

Object	Authorization required
Fila	MQZAO_CREATE **(“2” na página 188)**
Tópico	MQZAO_CREATE **(“2” na página 188)**
Processo	MQZAO_CREATE **(“2” na página 188)**
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CREATE **(“2” na página 188)**
Informações sobre Autenticação	MQZAO_CREATE **(“2” na página 188)**
Canal	MQZAO_CREATE **(“2” na página 188)**
Canal de conexão do cliente	MQZAO_CREATE **(“2” na página 188)**
Listener	MQZAO_CHANGE
Serviço	MQZAO_CHANGE

Criar *object* (com substituição) (“3” na página 188, “4” na página 188)

Object	Authorization required
Fila	MQZAO_CHANGE
Tópico	MQZAO_CHANGE
Processo	MQZAO_CHANGE
Gerenciador de filas	Não-aplicável
Lista de Nomes	MQZAO_CHANGE
Informações sobre Autenticação	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexão do cliente	MQZAO_CHANGE
Listener	MQZAO_CHANGE
Serviço	MQZAO_CHANGE

Excluir *object*

Object	Authorization required
Fila	MQZAO_DELETE
Tópico	MQZAO_DELETE
Processo	MQZAO_DELETE
Gerenciador de filas	MQZAO_DELETE
Lista de Nomes	MQZAO_DELETE
Informações sobre Autenticação	MQZAO_DELETE
Canal	MQZAO_DELETE

Object	Authorization required
Canal de conexão do cliente	MQZAO_DELETE
Listener	MQZAO_DELETE
Serviço	MQZAO_DELETE

Inquire *object*

Object	Authorization required
Fila	MQZAO_DISPLAY
Tópico	MQZAO_DISPLAY
Processo	MQZAO_DISPLAY
Gerenciador de filas	MQZAO_DISPLAY
Lista de Nomes	MQZAO_DISPLAY
Informações sobre Autenticação	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de conexão do cliente	MQZAO_DISPLAY
Listener	MQZAO_DISPLAY
Serviço	MQZAO_DISPLAY

Inquire *object* names

Object	Authorization required
Fila	Sem verificação
Tópico	Sem verificação
Processo	Sem verificação
Gerenciador de filas	Sem verificação
Lista de Nomes	Sem verificação
Informações sobre Autenticação	Sem verificação
Canal	Sem verificação
Canal de conexão do cliente	Sem verificação
Listener	Sem verificação
Serviço	Sem verificação

Executar ping no Canal

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável

Object	Authorization required
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

Redefinir Canal

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL_EXTENDED
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

Reconfigurar as Estatísticas de Fila

Object	Authorization required
Fila	MQZAO_DISPLAY e MQZAO_CHANGE
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	Não-aplicável
Canal de conexão do cliente	Não-aplicável
Listener	
Serviço	

Resolver Canal

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável

Object	Authorization required
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL_EXTENDED
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

Iniciar o Canal

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

Parar Canal

Object	Authorization required
Fila	Não-aplicável
Tópico	Não-aplicável
Processo	Não-aplicável
Gerenciador de filas	Não-aplicável
Lista de Nomes	Não-aplicável
Informações sobre Autenticação	Não-aplicável
Canal	MQZAO_CONTROL
Canal de conexão do cliente	Não-aplicável
Listener	Não-aplicável
Serviço	Não-aplicável

Nota:

1. Para comandos Copy, a autoridade MQZAO_DISPLAY também é necessária para o objeto De.

2. A autoridade MQZAO_CREATE não é específica a um determinado objeto ou tipo de objeto. A autoridade Criar é concedida para todos os objetos de um gerenciador de filas especificado, especificando-se um tipo de objeto de QMGR no comando GRTMQMAUT.
3. Para comandos de Criação, a autoridade MQZAO_DISPLAY também é necessária para o SYSTEM.DEFAULT.* objeto.
4. Essa opção se aplicará se o objeto a ser substituído já existir. Caso contrário, a verificação será como para Copy ou Create sem substituição.

IBM i Perfis genéricos do OAM no IBM i

Os perfis genéricos Object authority manager (OAM) permitem configurar a autoridade que um usuário tem com vários objetos de uma vez, em vez de ter que emitir comandos **GRTMQMAUT** separados em cada objeto individual ao ser criado. O uso de perfis genéricos no comando **GRTMQMAUT** permite configurar uma autoridade genérica para todos os futuros objetos criados que se ajustarem a esse perfil.

O restante desta seção descreve o uso de perfis genéricos em mais detalhes:

- "Utilizando Caracteres Curinga" na página 188
- "Prioridades do Perfil" na página 189

Utilizando Caracteres Curinga

O que torna um perfil genérico é o uso de caracteres especiais (caracteres curinga) no nome do perfil. Por exemplo, o caractere curinga de ponto de interrogação (?) corresponde a qualquer caractere único em um nome. Portanto, se você especificar ABC.?EF, a autorização fornecida a esse perfil se aplicará a qualquer objeto criado com os nomes ABC.DEF, ABC.CEF, ABC.BEF, etc.

Os caracteres curinga disponíveis são:

?

Use o ponto de interrogação (?) em vez de qualquer caractere. Por exemplo, AB.?D seria aplicado aos objetos AB.CD, AB.ED e AB.FD.

*****

Use o asterisco (*) como:

- Um *qualificador* em um nome de perfil para corresponder a qualquer qualificador em um nome de objeto. Um qualificador é a parte de um nome de objeto delimitado por um ponto. Por exemplo, em ABC.DEF.GHI, os qualificadores são ABC, DEF e GHI.

  Por exemplo, ABC.*.JKL seria aplicado aos objetos ABC.DEF.JKL e ABC.GHI.JKL. (Observe que ele **não** seria aplicado ao ABC.JKL; * usado nesse contexto sempre indica um qualificador.)
- Um caractere dentro de um qualificador em um nome de perfil para corresponder a zero ou mais caracteres dentro do qualificador em um nome de objeto.

  Por exemplo, ABC.DE*.JKL seria aplicado aos objetos ABC.DE.JKL, ABC.DEF.JKL e ABC.DEGH.JKL.

******

Use o asterisco duplo (**) ***uma vez*** em um nome de perfil como:

- O nome do perfil inteiro para corresponder a todos os nomes de objeto. Por exemplo, se você usar a palavra-chave OBJTYPE (*PRC) para identificar processos e, em seguida, usar ** como o nome do perfil, irá alterar as autorizações de todos os processos.
- Como o qualificador inicial, do meio ou final em um nome de perfil para corresponder a zero ou mais qualificadores em um nome de objeto. Por exemplo, **.ABC identifica todos os objetos com o qualificador final ABC.

Prioridades do Perfil

Um ponto importante a ser entendido ao usar perfis genéricos é a prioridade que os perfis recebem ao decidirem quais autoridades aplicar a um objeto que está sendo criado. Por exemplo, suponha que você emitiu estes comandos:

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

O primeiro fornece autoridade put para todas as filas para o principal FRED com nomes que correspondem ao perfil AB. *; O segundo fornece autoridade get para os mesmos tipos de fila que correspondem ao perfil AB.C*.

Suponha que agora você crie uma fila chamada AB.CD. De acordo com as regras para correspondência de curinga, GRTMQMAUT poderia se aplicar a essa fila. Portanto, ele tem autoridade put ou get?

Para localizar a resposta, você aplica a regra que, sempre que vários perfis puderem se aplicar a um objeto, **apenas o mais específico será aplicado**. A maneira de aplicação dessa regra é comparando-se os nomes dos perfis da esquerda para a direita. Sempre que forem diferentes, um caractere não genérico será mais específico do que um genérico. Portanto, no exemplo anterior, o AB.CD da fila **possui** autoridade (AB.C* é mais específico do que AB.*).

Ao comparar os caracteres genéricos, a ordem de *especificidade* é:

1. ?
2. *
3. **

IBM i Especificando o serviço de autorização instalado no IBM i

É possível especificar qual componente de serviço de autorização usar.

O parâmetro **`Service Component name`** on **GRTMQMAUT** e **RVKMQMAUT** permite especificar o nome do componente de serviço de autorização instalado.

Selecionar **F24** no painel inicial, seguido de **F9=Todos os parâmetros** no próximo painel de um dos comandos, permite especificar o componente de autorização instalado (*DFT) ou o nome do componente de serviço de autorização necessário especificado na sub-rotina de Serviço do arquivo qm.ini do gerenciador de filas.

DSPMQMAUT também possui esse parâmetro extra. Esse parâmetro permite procurar todos os componentes de autorização instalados (*DFT) ou o nome do componente de serviço de autorização especificado, para o nome do objeto, o tipo de objeto e o usuário especificados

IBM i Trabalhando com e sem perfis de autoridade em IBM i

Use estas informações para aprender como trabalhar com perfis de autoridade e como trabalhar sem perfis de autoridade.

É possível trabalhar com perfis de autoridade, conforme explicado em “Trabalhando com Perfis de Autoridade” na página 189, ou sem eles, conforme explicado aqui:

Para trabalhar sem perfis de autoridade, use *NONE como um parâmetro de Autoridade em **GRTMQMAUT** para criar perfis sem autoridade. Isso deixa os perfis existentes inalterados.

Em **RVKMQMAUT**, use *REMOVE como um parâmetro de Autoridade para remover um perfil de autoridade existente.

Trabalhando com Perfis de Autoridade

Há dois comandos associados à criação de perfil de autoridade:

- **WRKMQMAUT**

- **WRKMQMAUTD**

É possível acessar esses comandos diretamente da linha de comandos ou do painel WRKMQM:

1. Digitando o nome do gerenciador de filas e pressionando a tecla `Enter` para acessar o painel de resultados **WRKMQM**.
2. Selecionando `F23=More options` neste painel

A Opção 24 seleciona o painel de resultados para o **WRKMQMAUT** comando e a opção 25 seleciona o comando **WRKMQMAUTI** , que é usado com a camada de ligações SSL

WRKMQMAUT

Este comando permite que você trabalhe com os dados de autoridade retido na fila de autoridades.

Nota: Para executar esse comando, você deve ter as autoridades `*connect` e `*admdsp` ao gerenciador de filas. No entanto, para criar ou excluir um perfil, é necessária a autoridade QMQMADM.

Se você fornecer as informações como saída para a tela, uma lista de nomes de perfis de autoridade, juntamente com seus tipos, será exibida. Se você imprimir a saída, receberá uma lista detalhada de todos os dados de autoridade, dos usuários registrados e de suas autoridades.

Inserir um nome de objeto ou perfil nesse painel e pressionar ENTER o leva ao painel de resultados para **WRKMQMAUT**

Se você selecionar `4=Delete`, irá para um novo painel a partir do qual poderá confirmar que deseja excluir todos os nomes de usuário registrados para o nome do perfil de autoridade genérico especificado. Essa opção executa **RVKMQMAUT** com a opção *REMOVE para todos os usuários e aplica-se **apenas** a nomes de perfis genéricos.

Se você selecionar `12=Work with profile` , acesse o painel de resultados do comando **WRKMQMAUTD** , conforme explicado em "WRKMQMAUTD" na página 190.

WRKMQMAUTD

Esse comando permite exibir todos os usuários registrados com um determinado nome de perfil de autoridade e tipo de objeto. Para executar esse comando, você deve ter as autoridades `*connect` e `*admdsp` ao gerenciador de filas. No entanto, para conceder, executar, criar ou excluir um perfil, é necessária a autoridade QMQMADM.

Selecionar `F24=More keys` no painel de entrada inicial, seguido pela opção `F9=All Parameters` exibe o Nome do Componente de Serviço para **GRTMQMAUT** e **RVKMQMAUT**.

Nota: A chave `F11=Display Object Authorizations` alterna entre os seguintes tipos de autoridades:

- Autorizações de objetos
- Autorizações de contexto
- autorizações MQI

As opções na tela são:

2=Grant
Leva-o ao painel **GRTMQMAUT** para incluir nas autoridades atuais.

3=Revoke
Leva-o ao painel **RVKMQMAUT** para remover algumas das definições atuais

4=Delete
Leva-o a um painel que permite excluir os dados de autoridade de usuários especificados. Isso executa **RVKMQMAUT** com a opção *REMOVE.

5=Display
Leva-o ao comando **DSPMQMAUT** existente

F6=Create
Leva-o ao painel **GRTMQMAUT** que permite criar um registro de autoridade de perfil.

IBM i Diretrizes do gerenciador de autoridade de objeto no IBM i

Dicas e sugestões adicionais para usar o gerenciador de autoridade de objeto (OAM)

Limitar o Acesso a Operações Sigilosas

Algumas operações são sigilosas; limite-as aos usuários privilegiados. Por exemplo,

- Acessando algumas filas especiais, como filas de transmissão ou a fila de comandos SYSTEM.ADMIN.COMMAND.QUEUE
- Execução de programas que usam opções de contexto completas de MQI
- Criando e copiando filas de aplicativos

Diretórios do Gerenciador de Filas

Os diretórios e as bibliotecas que contêm filas e outros dados do gerenciador de filas são privativos ao produto. Não use comandos padrão do sistema operacional para conceder ou revogar autorizações para recursos do MQI.

Filas

A autoridade para uma fila dinâmica baseia-se naquela da fila modelo da qual é derivada, mas não é necessariamente a mesma.

Para filas de alias e filas remotas, a autorização é aquela do próprio objeto, não da fila à qual a fila do alias ou a fila remota é resolvida. É possível autorizar um perfil de usuário para acessar uma fila de alias que é resolvida para uma fila local à qual o perfil do usuário não possui permissões de acesso.

Limite a autoridade para criar filas a usuários privilegiados. Se você não fizer isso, os usuários podem ignorar o controle de acesso normal criando um alias.

Autoridade de Usuário Alternativo

A autoridade de usuário alternativo controla se um perfil do usuário pode usar a autoridade de outro perfil do usuário ao acessar um objeto do IBM MQ. Essa técnica é essencial onde um servidor recebe solicitações de um programa e deseja assegurar-se de que o programa possui a autoridade necessária para a solicitação. O servidor pode ter a autoridade necessária, mas precisa saber se o programa tem a autoridade para as ações solicitadas.

Por exemplo:

- Um programa do servidor em execução com um perfil de usuário PAYSERV recupera uma mensagem de solicitação de uma fila que foi colocada na fila pelo perfil de usuário USER1.
- Quando o programa do servidor recebe a mensagem de solicitação, ele processa a solicitação e coloca a resposta de volta na fila de resposta especificada com a mensagem de solicitação.
- Em vez de usar seu próprio perfil de usuário (PAYSERV) para autorizar a abertura da fila de resposta, o servidor pode especificar algum outro perfil de usuário, neste caso, USER1. Neste exemplo, é possível usar a autoridade de usuário alternativo para controlar se PAYSERV tem permissão para especificar USER1 como um perfil de usuário alternativo ao abrir a fila de resposta.

O perfil de usuário alternativo é especificado no campo *AlternateUserId* do descritor de objeto.

Nota: É possível usar perfis de usuário alternativo em qualquer objeto do IBM MQ. O uso de um perfil de usuário alternativo não afeta o perfil do usuário usado por nenhum outro gerenciador de recursos.

Autoridade de Contexto

Contexto são informações que se aplicam a uma determinada mensagem e está contido no descritor de mensagens, MQMD, que faz parte da mensagem.

Para obter as descrições dos campos do descritor de mensagens relacionados ao contexto, consulte MQMD-Descritor de mensagens

Para obter informações sobre as opções de contexto, consulte Contexto da mensagem.

Considerações de Segurança Remota

Para segurança remota, considere:

Autoridade de transmissão
Para segurança em gerenciadores de filas, é possível especificar a autoridade put usada quando um canal recebe uma mensagem enviada de outro gerenciador de filas.

Este parâmetro é válido somente para os tipos de canal RCVR, RQSTR ou CLUSRCVR. Especifique o atributo de canal PUTAUT da seguinte forma:

DEF
Perfil do usuário padrão. Este é o perfil do usuário QMQM sob o qual o agente do canal de mensagens está em execução.

CTX
O perfil do usuário no contexto da mensagem.

Filas de transmissão
Os gerenciadores de filas colocam mensagens remotas automaticamente em uma fila de transmissão; não é necessária nenhuma autoridade especial. No entanto, a colocação de uma mensagem diretamente em uma fila de transmissão exige autorização especial.

Saídas do canal
É possível usar saídas de canais para aumentar a segurança.

Registros de Autenticação de Canal
Use para exercer um controle mais preciso sobre o acesso concedido para conectar-se aos sistemas em um nível de canal.

Para obter mais informações sobre segurança remota, consulte “Autorização de canal” na página 120.

Protegendo canais com SSL/TLS

O protocolo Segurança da Camada de Transporte (TLS) fornece segurança de canal, com proteção contra espionagem do tráfego de rede, violação e personificação. O suporte do IBM MQ para TLS permite especificar, na definição de canal, que um determinado canal usa a segurança TLS. Também é possível especificar detalhes da segurança desejada, como o algoritmo de criptografia que você deseja usar.

O suporte de TLS no IBM MQ usa o *objeto de informações sobre autenticação* do gerenciador de filas e vários comandos de CL e MQSC e parâmetros de gerenciador de filas e de canal que definem o suporte do TLS requerido em detalhe.

Os comandos de CL a seguir suportam TLS:

WRKMQMAUTI
Trabalhar com os atributos de um objeto de informações sobre autenticação.

CHGMQMAUTI
Modificar os atributos de um objeto de informações sobre autenticação.

CRTMQMAUTI
Criar um objeto de informações sobre autenticação.

CPYMQMAUTI
Criar um objeto de informações sobre autenticação copiando um existente.

DLTMQMAUTI
Excluir um objeto de informações sobre autenticação.

DSPMQMAUTI
Exibe os atributos para um objeto de informações sobre autenticação específico.

Para obter uma visão geral de segurança do canal usando TLS, veja

- Protegendo canais com TLS

Para obter detalhes de comandos PCF associados ao TLS, veja

- Mudar, copiar e criar objeto de informações sobre autenticação
- Excluir Objeto de Informações sobre Autenticação
- Investigar Objeto de Informações sobre Autenticação

z/OS Setting up security on z/OS

Security considerations specific to z/OS.

Security in IBM MQ for z/OS is controlled using RACF or an equivalent external security manager (ESM).

The following instructions assume that you are using RACF.

Related concepts
Security scenario: two queue managers on z/OS
Security scenario: queue sharing group on z/OS

z/OS RACF security classes

RACF classes are used to hold the profiles required for IBM MQ security checking. Many of the member classes have equivalent group classes. You must activate the classes and enable them to accept generic profiles.

Each RACF class holds one or more profiles used at some point in the checking sequence, as shown in Table 23 on page 193.

Table 23. RACF classes used by IBM MQ

Member class	Group class	Contents
MQADMIN	GMQADMIN	Profiles that are used mainly for administrative functions. For example: • Profiles for IBM MQ security switches. • The RESLEVEL security profile. • Profiles for alternate user security. • Profiles for context security. • Profiles for command resource security. This class can hold only uppercase RACF profiles.

Table 23. RACF classes used by IBM MQ (continued)		
Member class	**Group class**	**Contents**
MXADMIN	GMXADMIN	Profiles that are used mainly for administrative functions. For example: • Profiles for IBM MQ security switches. • The RESLEVEL security profile. • Profiles for alternate user security. • Profiles for context security. • Profiles for command resource security. This class can hold both uppercase and mixed-case RACF profiles.
MQCONN		Profiles used for connection security.
MQCMDS		Profiles used for command security.
MQQUEUE	GMQQUEUE	Uppercase profiles used in queue resource security.
MXQUEUE	GMXQUEUE	Mixed-case and uppercase profiles used in queue resource security.
MQPROC	GMQPROC	Uppercase profiles used in process resource security.
MXPROC	GMXPROC	Mixed-case and uppercase profiles used in process resource security.
MQNLIST	GMQNLIST	Uppercase profiles used in namelist resource security.
MXNLIST	GMXNLIST	Mixed-case and uppercase profiles used in namelist resource security.
MXTOPIC	GMXTOPIC	Mixed-case and uppercase profiles used in topic security.

Some classes have a related *group class* that enables you to put together groups of resources that have similar access requirements. For details about the difference between the member and group classes and when to use a member or group class, see the z/OS Security Server RACF Security Administrator's Guide.

The classes must be activated before security checks can be made. To activate all the IBM MQ classes, you can use this RACF command:

```
SETROPTS CLASSACT(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                  MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

You should also ensure that you set up the classes so that they can accept generic profiles. You also do this with the RACF command **SETROPTS**, for example:

```
SETROPTS GENERIC(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                 MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

z/OS RACF profiles

All RACF profiles used by IBM MQ contain a prefix, which is either the queue manager name or the queue sharing group name. Be careful when you use the percent sign as a wildcard.

All RACF profiles used by IBM MQ contain a prefix. For queue sharing group level security, this is the queue sharing group name. For queue manager level security, the prefix is the queue manager name. If you are using a mixture of queue manager and queue sharing group level security, you will use profiles with both types of prefix. Queue sharing group and queue manager level security are described in Security controls and options in IBM MQ for z/OS.

For example, if you want to protect a queue called QUEUE_FOR_SUBSCRIBER_LIST in queue sharing group QSG1 at queue sharing group level, the appropriate profile would be defined to RACF as:

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

If you want to protect a queue called QUEUE_FOR_LOST_CARD_LIST, that belongs to queue manager STCD at queue manager level, the appropriate profile would be defined to RACF as:

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

This means that different queue managers and queue sharing groups can share the same RACF database and yet have different security options.

Do not use generic queue manager names in profiles to avoid unanticipated user access.

IBM MQ allows the use of the percent sign (%) in object names. However, RACF uses the % character as a single-character wildcard. This means that when you define an object name with a % character in its name, you must consider this when you define the corresponding profile.

For example, for the queue CREDIT_CARD_%_RATE_INQUIRY, on queue manager CRDP, the profile would be defined to RACF as follows:

```
RDEFINE MQQUEUE CRDP.CREDIT_CARD_%_RATE_INQUIRY
```

This queue cannot be protected by a generic profile, such as, CRDP.**.

IBM MQ allows the use of mixed-case characters in object names. You can protect these objects by defining:

1. Mixed-case profiles in the appropriate mixed-case RACF classes, or
2. Generic profiles in the appropriate uppercase RACF classes.

To use mixed-case profiles and mixed-case RACF classes you must follow the steps described in “Migrating a z/OS queue manager to mixed-case security” on page 272.

There are some profiles, or parts of profiles, that remain uppercase only as the values are provided by IBM MQ. These are:

- Switch profiles.
- All high-level qualifiers (HLQ) including subsystem and queue sharing group identifiers.
- Profiles for SYSTEM objects.
- Profiles for Default objects.
- The **MQCMDS** class, so all command profiles are uppercase only.
- The **MQCONN** class, so all connection profiles are uppercase only.
- **RESLEVEL** profiles.
- The `'object'` qualification in command resource profiles; for example, `hlq.QUEUE.queuename`. The resource name only is mixed case.
- Dynamic queue profiles `hlq.CSQOREXX.*`, `hlq.CSQUTIL.*`, and `CSQXCMD.*`.
- The `'CONTEXT'` part of `hlq.CONTEXT.resourcename`.
- The `'ALTERNATE.USER'` part of `hlq.ALTERNATE.USER.userid`.

For example, you can define a profile to grant access to a queue called PAYROLL.Dept1 on queue manager QM01 in one of the following ways.

- If you are using mixed-case profiles, you can define a profile in the IBM MQ RACF class MXQUEUE using the following command:

```
RDEFINE MXQUEUE MQ01.PAYROLL.Dept1
```

- If you are using uppercase profiles, you can define a profile in the IBM MQ RACF class MQQUEUE using the following command:

```
RDEFINE MQQUEUE MQ01.PAYROLL.*
```

The first example, using mixed-case profiles, gives you more granular control over granting authority to access the resource.

z/OS Switch profiles

To control the security checking performed by IBM MQ, you use *switch profiles*. A switch profile is a normal RACF profile that has a special meaning to IBM MQ. The access list in switch profiles is not used by IBM MQ.

IBM MQ maintains an internal switch for each switch type shown in tables Switch profiles for subsystem level security, Switch profiles for queue sharing group or queue manager level security,and Switch profiles for resource checking. Switch profiles can be maintained at queue sharing group level, or at queue manager level, or at a combination of both. Using a single set of queue sharing group security switch profiles, you can control security on all the queue managers within a queue sharing group.

When a security switch is set on, the security checks associated with the switch are performed. When a security switch is set off, the security checks associated with the switch are bypassed. The default is that all security switches are set on.

z/OS *Switches and classes*

When you start a queue manager or refresh security, IBM MQ sets switches according to the state of various RACF classes.

When a queue manager is started (or when the MQADMIN or MXADMIN class is refreshed by the IBM MQ REFRESH SECURITY command), IBM MQ first checks the status of RACF and the appropriate class:

- The MQADMIN class if you are using uppercase profiles
- The MXADMIN class if you are using mixed case profile.

It sets the subsystem security switch off if any of these conditions is true:

- RACF is inactive or not installed.
- The MQADMIN or MXADMIN class is not defined (these classes are always defined for RACF because they are included in the class descriptor table (CDT)).
- The MQADMIN or MXADMIN class has not been activated.

If both RACF and the MQADMIN or MXADMIN class are active, IBM MQ checks the MQADMIN or MXADMIN class to see whether any of the switch profiles have been defined. It first checks the profiles described in “Profiles to control subsystem security” on page 197. If subsystem security is not required, IBM MQ sets the internal subsystem security switch off, and performs no further checks.

The profiles determine whether the corresponding IBM MQ switch is set on or off.

- If the switch is off, that type of security is deactivated.
- If any IBM MQ switch is set on, IBM MQ checks the status of the RACF class associated with the type of security corresponding to the IBM MQ switch. If the class is not installed or not active, the IBM MQ switch is set off. For example, process security checks are not carried out if the MQPROC or MXPROC

class has not been activated. The class not being active is equivalent to defining NO.PROCESS.CHECKS profile for every queue manager and queue sharing group that uses this RACF database.

z/OS *How switches work*

To set off a security switch, define a NO.* switch profile for it. You can override a NO.* profile set at the queue sharing group level by defining a YES.* profile for a queue manager.

To set off a security switch, you need to define a NO.* switch profile for it. The existence of a NO.* profile means that security checks are **not** performed for that type of resource, unless you choose to override a queue sharing group level setting on a particular queue manager. This is described in "Overriding queue sharing group level settings" on page 197.

If your queue manager is not a member of a queue sharing group, you do not need to define any queue sharing group level profiles or any override profiles. However, you must remember to define these profiles if the queue manager joins a queue sharing group at a later date.

Each NO.* switch profile that IBM MQ detects turns off the checking for that type of resource. Switch profiles are activated during startup of the queue manager. If you change the switch profiles while any affected queue managers are running, you can get IBM MQ to recognize the changes by issuing the IBM MQ REFRESH SECURITY command.

The switch profiles must always be defined in the MQADMIN or MXADMIN class. Do not define them in the GMQADMIN or GMXADMIN class. Tables Switch profiles for subsystem level security and Switch profiles for resource checking show the valid switch profiles and the security type they control.

Overriding queue sharing group level settings

You can override queue sharing group level security settings for a particular queue manager that is a member of that group. If you want to perform queue manager checks on an individual queue manager that are not performed on other queue managers in the group, use the (qmgr-name.YES.*) switch profiles.

Conversely, if you do not want to perform a certain check on one particular queue manager within a queue sharing group, define a (qmgr-name.NO.*) profile for that particular resource type on the queue manager, and do not define a profile for the queue sharing group. (IBM MQ only checks for a queue sharing group level profile if it does not find a queue manager level profile.)

z/OS *Profiles to control subsystem security*

IBM MQ checks whether subsystem security checks are required for the subsystem, for the queue manager, and for the queue sharing group.

The first security check made by IBM MQ is used to determine whether security checks are required for the whole IBM MQ subsystem. If you specify that you do not want subsystem security, no further checks are made.

The following switch profiles are checked to determine whether subsystem security is required. Figure 14 on page 198 shows the order in which they are checked.

Table 24. Switch profiles for subsystem level security

Switch profile name	Type of resource or checking that is controlled
qmgr-name.NO.SUBSYS.SECURITY	Subsystem security for this queue manager
qsg-name.NO.SUBSYS.SECURITY	Subsystem security for this queue sharing group
qmgr-name.YES.SUBSYS.SECURITY	Subsystem security override for this queue manager

If your queue manager is not a member of a queue sharing group, IBM MQ checks for the qmgr-name.NO.SUBSYS.SECURITY switch profile only.

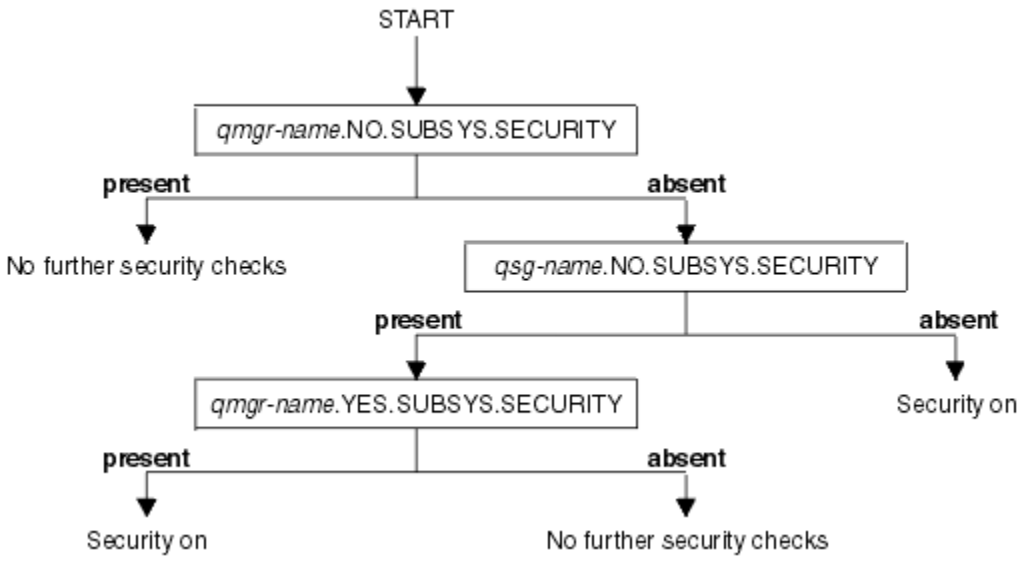


Figure 14. Checking for subsystem security

z/OS *Profiles to control queue sharing group or queue manager level security*

If subsystem security checking is required, IBM MQ checks whether security checking is required at queue sharing group or queue manager level.

When IBM MQ has determined that security checking is required, it then determines whether checking is required at queue sharing group or queue manager level, or both. These checks are not performed if your queue manager is not a member of a queue sharing group.

The following switch profiles are checked to determine the level required. Figure 15 on page 199 and Figure 16 on page 199 show the order in which they are checked.

Table 25. Switch profiles for queue sharing group or queue manager level security

Switch profile name	Type of resource or checking that is controlled
qmgr-name.NO.QMGR.CHECKS	No queue manager level checks for this queue manager
qsg-name.NO.QMGR.CHECKS	No queue manager level checks for this queue sharing group
qmgr-name.YES.QMGR.CHECKS	Queue manager level checks override for this queue manager
qmgr-name.NO.QSG.CHECKS	No queue sharing group level checks for this queue manager
qsg-name.NO.QSG.CHECKS	No queue sharing group level checks for this queue sharing group
qmgr-name.YES.QSG.CHECKS	Queue sharing group level checks override for this queue manager

If subsystem security is active, you cannot switch off both queue sharing group and queue manager level security. If you try to do so, IBM MQ sets security checking on at both levels.

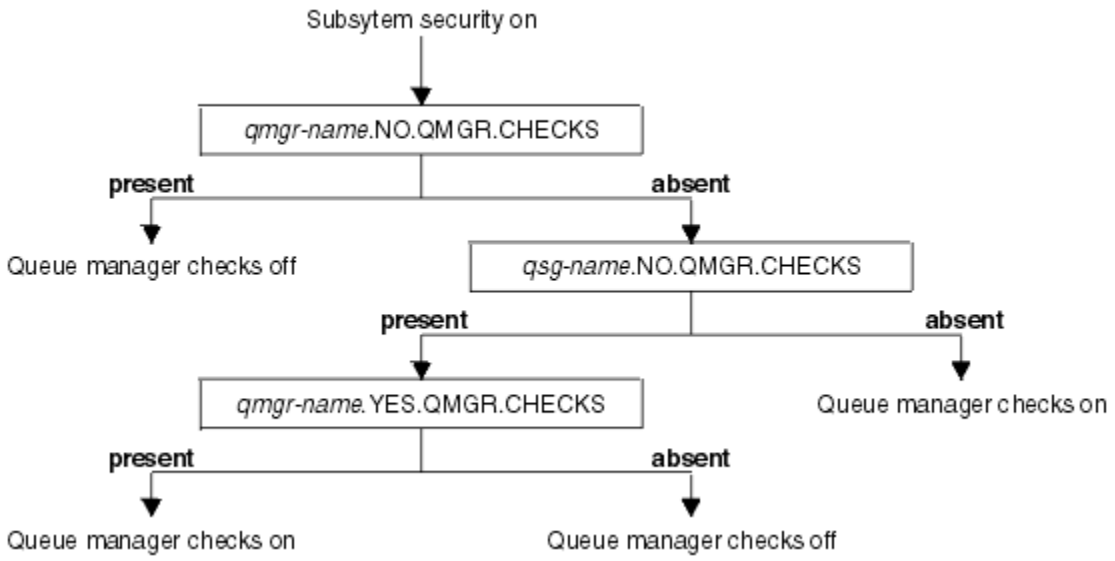


Figure 15. Checking for queue manager level security

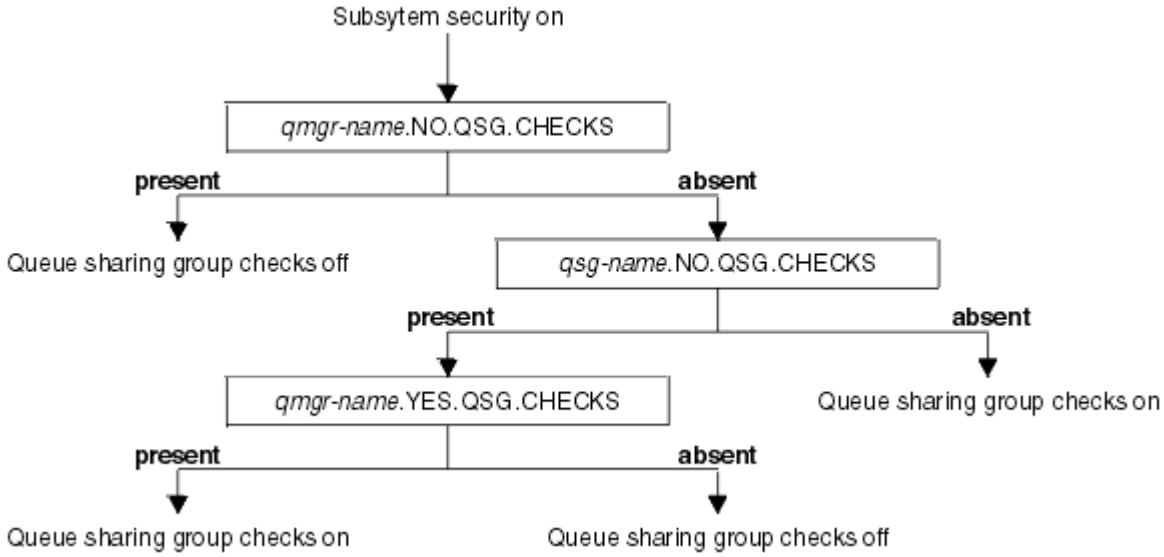


Figure 16. Checking for queue sharing group level security

z/OS *Valid combinations of security switches*

Only certain combinations of switches are valid. If you use a combination of switch settings that is not valid, message CSQH026I is issued and security checking is set on at both queue sharing group and queue manager level.

Table 26 on page 199, Table 27 on page 200, Table 28 on page 200, and Table 29 on page 200 show the sets of combinations of switch settings that are valid for each type of security level.

Table 26. Valid security switch combinations for queue manager level security

Combinations
qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QSG.CHECKS

Table 26. Valid security switch combinations for queue manager level security (continued)
Combinations
qmgr-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS
qsg-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS

Table 27. Valid security switch combinations for queue sharing group level security
Combinations
qmgr-name.NO.QMGR.CHECKS
qsg-name.NO.QMGR.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS

Table 28. Valid security switch combinations for queue manager and queue sharing group level security
Combinations
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS No QSG.* profiles defined
No QMGR.* profiles defined qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS
No profiles for either switch defined

*Table 29. Other valid security switch combinations that switch both levels of checking **on**.*
Combinations
qmgr-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS

*Table 29. Other valid security switch combinations that switch both levels of checking **on**. (continued)*
Combinations
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS

z/OS *Resource level checks*

A number of switch profiles are used to control access to resources. Some stop checking being performed on either a queue manager or a queue sharing group. These can be overridden by profiles that enable checking for specific queue managers.

Table 30 on page 201 shows the switch profiles used to control access to IBM MQ resources.

If your queue manager is part of a queue sharing group and you have both queue manager and queue sharing group security active, you can use a YES.* switch profile to override queue sharing group level profiles and specifically turn on security for a particular queue manager.

Some profiles apply to both queue managers and queue sharing groups. These are prefixed by the string *hlq* and you should substitute the name of your queue sharing group or queue manager, as applicable. Profile names shown prefixed by *qmgr-name* are queue manager override profiles; you should substitute the name of your queue manager.

Table 30. Switch profiles for resource checking

Type of resource checking that is controlled	**Switch profile name**	**Override profile for a particular queue manager**
Connection security	hlq.NO.CONNECT.CHECKS	qmgr-name.YES.CONNECT.CHECKS
Queue security	hlq.NO.QUEUE.CHECKS	qmgr-name.YES.QUEUE.CHECKS
Process security	hlq.NO.PROCESS.CHECKS	qmgr-name.YES.PROCESS.CHECKS
Namelist security	hlq.NO.NLIST.CHECKS	qmgr-name.YES.NLIST.CHECKS
Context security	hlq.NO.CONTEXT.CHECKS	qmgr-name.YES.CONTEXT.CHECKS
Alternate user security	hlq.NO.ALTERNATE.USER.CHECKS	qmgr-name.YES.ALTERNATE.USER.CHECKS
Command security	hlq.NO.CMD.CHECKS	qmgr-name.YES.CMD.CHECKS
Command resource security	hlq.NO.CMD.RESC.CHECKS	qmgr-name.YES.CMD.RESC.CHECKS
Topic security	hlq.NO.TOPIC.CHECKS	qmgr-name.YES.TOPIC.CHECKS
Note: Generic switch profiles such as hlq.NO.** are ignored by IBM MQ		

For example, if you want to perform process security checks on queue manager QM01, which is a member of queue sharing group QSG3 but you do not want to perform process security checks on any of the other queue managers in the group, define the following switch profiles:

```
QSG3.NO.PROCESS.CHECKS
QM01.YES.PROCESS.CHECKS
```

If you want to have queue security checks performed on all the queue managers in the queue sharing group, except QM02, define the following switch profile:

```
QM02.NO.QUEUE.CHECKS
```

(There is no need to define a profile for the queue sharing group because the checks are automatically enabled if there is no profile defined.)

z/OS *An example of defining switches*

Different IBM MQ subsystems have different security requirements, which can be implemented using different switch profiles.

Four IBM MQ subsystems have been defined:

- MQP1 (a production system)
- MQP2 (a production system)
- MQD1 (a development system)
- MQT1 (a test system)

All four queue managers are members of queue sharing group QS01. All IBM MQ RACF classes have been defined and activated.

These subsystems have different security requirements:

- The production systems require full IBM MQ security checking to be active at queue sharing group level on both systems.

  This is done by specifying the following profile:

  ```
  RDEFINE MQADMIN QS01.NO.QMGR.CHECKS
  ```

  This sets queue sharing group level checking for all the queue managers in the queue sharing group. You do not need to define any other switch profiles for the production queue managers because you want to check everything for these systems.
- Test queue manager MQT1 also requires full security checking. However, because you might want to change this later, security can be defined at queue manager level so that you can change the security settings for this queue manager without affecting the other members of the queue sharing group.

  This is done by defining the NO.QSG.CHECKS profile for MQT1 as follows:

  ```
  RDEFINE MQADMIN MQT1.NO.QSG.CHECKS
  ```
- Development queue manager MQD1 has different security requirements from the rest of the queue sharing group. It requires only connection and queue security to be active.

  This is done by defining a `MQD1.YES.QMGR.CHECKS` profile for this queue manager, and then defining the following profiles to switch off security checking for the resources that do not need to be checked:

  ```
  RDEFINE MQADMIN MQD1.NO.CMD.CHECKS
  RDEFINE MQADMIN MQD1.NO.CMD.RESC.CHECKS
  RDEFINE MQADMIN MQD1.NO.PROCESS.CHECKS
  RDEFINE MQADMIN MQD1.NO.NLIST.CHECKS
  RDEFINE MQADMIN MQD1.NO.CONTEXT.CHECKS
  RDEFINE MQADMIN MQD1.NO.ALTERNATE.USER.CHECKS
  ```

When the queue manager is active, you can display the current security settings by issuing the DISPLAY SECURITY MQSC command.

You can also change the switch settings when the queue manager is running by defining or deleting the appropriate switch profile in the MQADMIN class. To make the changes to the switch settings active, you must issue the REFRESH SECURITY command for the MQADMIN class.

See “Refreshing queue manager security on z/OS” on page 254 for more details about using the DISPLAY SECURITY and REFRESH SECURITY commands.

z/OS Profiles used to control access to IBM MQ resources

You must define RACF profiles to control access to IBM MQ resources, in addition to the switch profiles that might have been defined. This collection of topics contains information about the RACF profiles for the different types of IBM MQ resource.

If you do not have a resource profile defined for a particular security check, and a user issues a request that would involve making that check, IBM MQ denies access. You do not have to define profiles for security types relating to any security switches that you have deactivated.

z/OS *Profiles for connection security*

If connection security is active, you must define profiles in the MQCONN class and permit the necessary groups or user IDs access to those profiles, so that they can connect to IBM MQ.

To enable a connection to be made, you must grant users RACF READ access to the appropriate profile. (If no queue manager level profile exists, and your queue manager is a member of a queue sharing group, checks might be made against queue sharing group level profiles, if the security is set up to do this.)

A connection profile qualified with a queue manager name controls access to a specific queue manager and users given access to this profile can connect to that queue manager. A connection profile qualified with queue sharing group name controls access to all queue managers within the queue sharing group for that connection type. For example, a user with access to `QS01.BATCH` can use a batch connection to any queue manager in queue sharing group QS01 that has not got a queue manager level profile defined.

Note:

1. For information about the user IDs checked for different security requests, see “User IDs for security checking on z/OS” on page 243.
2. Resource level security (RESLEVEL) checks are also made at connection time. For details, see “O perfil de segurança RESLEVEL” on page 237.

IBM MQ security recognizes the following different types of connection:

- Batch (and batch-type) connections, these include:
  - z/OS batch jobs
  - TSO applications
  - z/OS UNIX System Services sign-ons
  - Db2 stored procedures
- CICS connections
- IMS connections from control and application processing regions
- The IBM MQ channel initiator

z/OS *Connection security profiles for batch connections*

Profiles for checking batch-type connections are composed of the queue manager or queue sharing group name followed by the word *BATCH*. Give the user ID associated with the connecting address space READ access to the connection profile.

Profiles for checking batch and batch-type connections take the form:

```
hlq.BATCH
```

where `hlq` can be either the `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails.

For batch or batch-type connection requests, you must permit the user ID associated with the connecting address space to access the connection profile. For example, the following RACF command allows users in the CONNTQM1 group to connect to the queue manager TQM1; these user IDs will be permitted to use any batch or batch-type connection.

```
RDEFINE MQCONN TQM1.BATCH UACC(NONE)
PERMIT TQM1.BATCH CLASS(MQCONN) ID(CONNTQM1) ACCESS(READ)
```

z/OS *Using **CHCKLOCL** on locally bound applications*

CHCKLOCL only applies to connections that are made through BATCH connections and does not apply to connections made from CICS or IMS. Connections made through the channel initiator are controlled by **CHCKCLNT**.

Overview

If you want to configure your z/OS queue manager to mandate user ID and password checking for some, but not all, of your locally bound applications, you need to do some additional configuration.

The reason for this is that once **CHCKLOCL** (*REQUIRED*) is configured, legacy batch applications that use the MQCONN API call can no longer connect to the queue manager.

For z/OS only, a more granular mechanism based on the connection security of an address space can be used to downgrade the global CHCKLOCL(REQUIRED) configuration to CHCKLOCL(OPTIONAL) for specifically defined user IDs. The mechanism used, is described in the following text, together with an example.

In order to allow more granularity on **CHCKLOCL** (*REQUIRED*) than just EVERYONE, you modify **CHCKLOCL** in the same manner as you modify the access level of the user ID associated with the connecting address space to the `hlq.batch` connection profiles in the MQCONN class.

If the address space user ID only has READ access, which is the minimum you require to be able to connect at all, the **CHCKLOCL** configuration applies as written.

If the address space user ID has UPDATE access (or above) then the **CHCKLOCL** configuration operates in *OPTIONAL* mode. That is, you do not have to provide a user ID and password, but if you do, the user ID and password must be a valid pair.

Connection security already configured for your z/OS queue manager

If you have connection security configured for your z/OS queue manager and you want **CHCKLOCL** (*REQUIRED*) to apply to WAS locally bound applications, and no others, carry out the following steps:

1. Start with **CHCKLOCL** (*OPTIONAL*) as your configuration. This means that any user ID and passwords that are supplied are checked for validity, but not mandated.
2. List all the users that have access to the connection security profiles by issuing the command:

   ```
   RLIST MQCONN MQ23.BATCH AUTHUSER
   ```

   This command displays, for example:

   ```
   CLASS    NAME
   -----    ----
   MQCONN   MQ23.BATCH

   USER     ACCESS  ACCESS COUNT
   ----     ------  ------ -----
   ```

```
JOHNDOE  READ    000009
JDOE1    READ    000003
WASUSER  READ    000000
```

3. For each user ID listed as having READ access, change the access to

```
UPDATE:- PERMIT MQ23.BATCH CLASS(MQCONN) ID(JOHNDOE) ACCESS(UPDATE)
```

4. Update the IBM MQ configuration to **CHCKLOCL** (*REQUIRED*).

   The combination of UPDATE access to MQ23.BATCH and the current setting means that you are using **CHCKLOCL** (*OPTIONAL*).

5. Now, apply the **CHCKLOCL** (*REQUIRED*) behavior to one specific user ID, for example WASUSER, so that all the connections coming from that region must provide a user ID and password.

   Do this by reversing the change you made previously, by issuing the command:

```
PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
```

Connection security is not configured for your z/OS queue manager

In this situation, you must:

1. Create connection profiles for hlq.BATCH in the MQCONN class, by issuing the command:

```
RDEFINE MQCONN MQ23.BATCH UACC(NONE)
```

2. Authorize all user IDs that create batch connections to the queue manager, so that they have UPDATE access to this profile. Doing this bypasses the **CHCKLOCL** (*REQUIRED*) requirement for the user ID and password at the time of connection.

   Do this by issuing the command:

```
PERMIT MQ23.BATCH CLASS(MQCONN)ID(JOHNDOE) ACCESS(UPDATE)
```

   These include user IDs:

   a. Used for CSQUTIL, ISPF panels, and other locally bound tools.

   b. Associated with batch like connections to the queue manager. Consider for example, Advanced Message Security, IBM Integration Bus, Db2 stored procedures, z/OS UNIX System Services and TSO users, and Java applications

3. Delete the switch profile for the queue manager by issuing the command:

```
hlq.NO.CONNECT.CHECKS
```

4. Now, apply the **CHCKLOCL** (*REQUIRED*) behavior to one specific user ID, for example WASUSER, so that all the connections coming from that region must provide a user ID and password.

   Do this by reversing the change you made previously, by issuing the command:

```
PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
```

z/OS *Connection security profiles for CICS connections*

Profiles for checking CICS connections are composed of the queue manager or queue sharing group name followed by the word *CICS* . Give the user ID associated with the CICS address space READ access to the connection profile.

Profiles for checking connections from CICS take the form:

```
hlq.CICS
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by CICS, you need only permit the CICS address space user ID access to the connection profile.

For example, the following RACF commands allow the CICS address space user ID KCBCICS to connect to the queue manager TQM1:

```
RDEFINE MQCONN TQM1.CICS UACC(NONE)
PERMIT TQM1.CICS CLASS(MQCONN) ID(KCBCICS) ACCESS(READ)
```

z/OS *Connection security profiles for IMS connections*

Profiles for checking IMS connections are composed of the queue manager or queue sharing group name followed by the word *IMS* . Give the IMS control and dependent region user IDs READ access to the connection profile.

Profiles for checking connections from IMS take the form:

```
hlq.IMS
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by IMS, permit access to the connection profile for the IMS control and dependent region user IDs.

For example, the following RACF commands allow:

- The IMS region user ID, IMSREG, to connect to the queue manager TQM1.
- Users in group BMPGRP to submit BMP jobs.

```
RDEFINE MQCONN TQM1.IMS UACC(NONE)
PERMIT TQM1.IMS CLASS(MQCONN) ID(IMSREG,BMPGRP) ACCESS(READ)
```

z/OS *Connection security profiles for the channel initiator*

Profiles for checking connections from the channel initiator are composed of the queue manager or queue sharing group name followed by the word *CHIN*. Give the user ID used by the channel initiator started task address space READ access to the connection profile.

Profiles for checking connections from the channel initiator take the form:

```
hlq.CHIN
```

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue sharing group name). If you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name. If it fails to find either profile, the connection request fails

For connection requests by the channel initiator, define access to the connection profile for the user ID used by the channel initiator started task address space.

For example, the following RACF commands allow the channel initiator address space running with user ID DQCTRL to connect to the queue manager TQM1:

```
RDEFINE MQCONN TQM1.CHIN UACC(NONE)
PERMIT TQM1.CHIN CLASS(MQCONN) ID(DQCTRL) ACCESS(READ)
```

z/OS *Profiles for queue security*

If queue security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to these profiles. Queue security profiles are named after the queue manager or queue sharing group, and the queue to be opened.

If queue security is active, you must:

- Define profiles in the **MQQUEUE** or **GMQQUEUE** classes if using uppercase profiles.
- Define profiles in the **MXQUEUE** or **GMXQUEUE** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use queues.

Profiles for queue security take the form:

```
hlq.queuename
```

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue sharing group name), and queuename is the name of the queue being opened, as specified in the object descriptor on the MQOPEN or MQPUT1 call.

A profile prefixed by the queue manager name controls access to a single queue on that queue manager. A profile prefixed by the queue sharing group name controls access to access to one or more queues with that queue name on all queue managers within the queue sharing group, or access to a shared queue by any queue manager within the group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that queue on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

If you are using shared queues, you are recommended to use queue sharing group level security.

For details of how queue security operates when the queue name is that of an alias or a model queue, see “Considerations for alias queues” on page 209 and “Considerations for model queues” on page 210 .

The RACF access required to open a queue depends on the MQOPEN or MQPUT1 options specified. If more than one of the MQOO_* and MQPMO_* options is coded, the queue security check is performed for the highest RACF authority required.

Table 31. Access levels for queue security using the MQOPEN or MQPUT1 calls	
MQOPEN or MQPUT1 option	**RACF access level required to hlq.queuename**
MQOO_BROWSE	READ
MQOO_INQUIRE	READ
MQOO_BIND_*	UPDATE
MQOO_INPUT_*	UPDATE
MQOO_OUTPUT or MQPUT1	UPDATE
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	UPDATE
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	UPDATE
MQOO_SAVE_ALL_CONTEXT	UPDATE
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	UPDATE
MQOO_SET	ALTER

For example, on IBM MQ queue manager QM77, all user IDs in the RACF group PAYGRP are to be given access to get messages from or put messages to all queues with names beginning with 'PAY.'. You can do this using these RACF commands:

```
RDEFINE MQQUEUE QM77.PAY.** UACC(NONE)
PERMIT QM77.PAY.** CLASS(MQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Also, all user IDs in the PAYGRP group must have access to put messages on queues that do not follow the PAY naming convention. For example:

```
REQUEST_QUEUE_FOR_PAYROLL
SALARY.INCREASE.SERVER
REPLIES.FROM.SALARY.MODEL
```

You can do this by defining profiles for these queues in the GMQQUEUE class and giving access to that class as follows:

```
RDEFINE GMQQUEUE PAYROLL.EXTRAS UACC(NONE)
        ADDMEM(QM77.REQUEST_QUEUE_FOR_PAYROLL,
               QM77.SALARY.INCREASE.SERVER,
               QM77.REPLIES.FROM.SALARY.MODEL)
PERMIT PAYROLL.EXTRAS CLASS(GMQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Note:

1. If the RACF access level that an application has to a queue security profile is changed, the changes only take effect for any new object handles obtained (that is, new MQOPEN s) for that queue. Those handles already in existence at the time of the change retain their existing access to the queue. If

an application is required to use its changed access level to the queue rather than its existing access level, it must close and reopen the queue for each object handle that requires the change.

2. In the example, the queue manager name QM77 could also be the name of a queue sharing group.

Other types of security checks might also occur at the time the queue is opened depending on the open options specified and the types of security that are active. See also "Profiles for context security" on page 223 and "Profiles for alternate user security" on page 221. For a summary table showing the open options and the security authorization needed when queue, context, and alternate user security are all active, see Table 36 on page 214.

If you are using publish/subscribe you must consider the following. When an MQSUB request is processed a security check is performed to ensure that the user ID making the request has the required access to put messages to the target IBM MQ queue as well as the required access to subscribe to the IBM MQ topic.

Table 32. Access levels for queue security using the MQSUB call

MQSUB option	**RACF access level required to hlq.queuename**
MQSO_ALTER, MQSO_CREATE, and MQSO_RESUME	UPDATE

Note:

1. The `hlq.queuename` is the destination queue for publications. When this is a managed queue, you need access to the appropriate model queue to be used for the managed queue and the dynamic queue that are created.
2. You can use a technique like this for the destination queue you provide on an MQSUB API call if you want to distinguish between the users making the subscriptions, and the users retrieving the publications from the destination queue.

z/OS *Considerations for alias queues*

When you issue an MQOPEN or MQPUT1 call for an alias queue, IBM MQ makes a resource check against the queue name specified in the object descriptor (MQOD) on the call. It does not check if the user is allowed access to the target queue name.

For example, an alias queue called PAYROLL.REQUEST resolves to a target queue of PAY.REQUEST. If queue security is active, you need only be authorized to access the queue PAYROLL.REQUEST. No check is made to see if you are authorized to access the queue PAY.REQUEST.

z/OS *Using alias queues to distinguish between MQGET and MQPUT requests*

The range of MQI calls available in one access level can cause a problem if you want to restrict access to a queue to allow only the **MQPUT** call or only the **MQGET** call. A queue can be protected by defining two aliases that resolve to that queue: one that enables applications to get messages from the queue, and one that enable applications to put messages on the queue.

The following text gives you an example of how you can define your queues to IBM MQ:

```
DEFINE QLOCAL(MUST_USE_ALIAS_TO_ACCESS) GET(ENABLED)
       PUT(ENABLED)

DEFINE QALIAS(USE_THIS_ONE_FOR_GETS) GET(ENABLED)
       PUT(DISABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)

DEFINE QALIAS(USE_THIS_ONE_FOR_PUTS) GET(DISABLED)
       PUT(ENABLED) TARGET(MUST_USE_ALIAS_TO_ACCESS)
```

You must also make the following RACF definitions:

```
RDEFINE MQQUEUE hlq.MUST_USE_ALIAS_TO_ACCESS UACC(NONE)
```

```
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_GETS UACC(NONE)
RDEFINE MQQUEUE hlq.USE_THIS_ONE_FOR_PUTS UACC(NONE)
```

Then you ensure that no users have access to the queue hlq.MUST_USE_ALIAS_TO_ACCESS, and give the appropriate users or groups access to the alias. You can do this using the following RACF commands:

```
PERMIT hlq.USE_THIS_ONE_FOR_GETS CLASS(MQQUEUE)
       ID(GETUSER,GETGRP) ACCESS(UPDATE)
PERMIT hlq.USE_THIS_ONE_FOR_PUTS CLASS(MQQUEUE)
       ID(PUTUSER,PUTGRP) ACCESS(UPDATE)
```

This means user ID GETUSER and user IDs in the group GETGRP are only allowed to get messages on MUST_USE_ALIAS_TO_ACCESS through the alias queue USE_THIS_ONE_FOR_GETS; and user ID PUTUSER and user IDs in the group PUTGRP are only allowed to put messages through the alias queue USE_THIS_ONE_FOR_PUTS.

Note:

1. If you want to use a technique like this, you must inform your application developers, so that they can design their programs appropriately.
2. You can use a technique like this for the destination queue you provide on an MQSUB API request if you want to distinguish between the users making the subscriptions and the users 'getting' the publications from the destination queue.

z/OS *Considerations for model queues*

To open a model queue, you must be able to open both the model queue itself and the dynamic queue to which it resolves. Define generic RACF profiles for dynamic queues, including dynamic queues used by IBM MQ utilities.

When you open a model queue, IBM MQ security makes two queue security checks:

1. Are you authorized to access the model queue?
2. Are you authorized to access the dynamic queue to which the model queue resolves?

If the dynamic queue name contains a trailing asterisk (*) character, this * is replaced by a character string generated by IBM MQ, to create a dynamic queue with a unique name. However, because the whole name, including this generated string, is used for checking authority, you should define generic profiles for these queues.

For example, an MQOPEN call uses a model queue name of CREDIT.CHECK.REPLY.MODEL and a dynamic queue name of CREDIT.REPLY.* on queue manager (or queue sharing group) MQSP.

To do this, you must issue the following RACF commands to define the necessary queue profiles:

```
RDEFINE MQQUEUE MQSP.CREDIT.CHECK.REPLY.MODEL
RDEFINE MQQUEUE MQSP.CREDIT.REPLY.**
```

You must also issue the corresponding RACF PERMIT commands to allow the user access to these profiles.

A typical dynamic queue name created by an MQOPEN is something like CREDIT.REPLY.A346EF00367849A0. The precise value of the last qualifier is unpredictable; this is why you should use generic profiles for such queue names.

A number of IBM MQ utilities put messages on dynamic queues. You should define profiles for the following dynamic queue names, and provide RACF UPDATE access to the relevant user IDs (see “User IDs for security checking on z/OS” on page 243 for the correct user IDs):

```
SYSTEM.CSQUTIL.*  (used by CSQUTIL)
SYSTEM.CSQOREXX.* (used by the operations and control panels)
SYSTEM.CSQXCMD.*  (used by the channel initiator when processing CSQINPX)
CSQ4SAMP.*        (used by the IBM MQ supplied samples)
```

You might also consider defining a profile to control use of the dynamic queue name used by default in the application programming copy members. The IBM MQ-supplied copybooks contain a default *DynamicQName*, which is CSQ.*. This enables an appropriate RACF profile to be established.

Note: Do not allow application programmers to specify a single * for the dynamic queue name. If you do, you must define an hlq.** profile in the MQQUEUE class, and you would have to give it wide-ranging access. This means that this profile could also be used for other non-dynamic queues that do not have a more specific RACF profile. Your users could, therefore, gain access to queues you do not want them to access.

z/OS *Close options on permanent dynamic queues*

If an application opens a permanent dynamic queue that was created by another application and then attempts to delete that queue with an MQCLOSE option, some extra security checks are applied when the attempt is made.

Table 33. Access levels for close options on permanent dynamic queues

MQCLOSE option	RACF access level required to hlq.queuename
MQCO_DELETE	ALTER
MQCO_DELETE_PURGE	ALTER

z/OS *Security and remote queues*

When a message is put on a remote queue, the queue security that is implemented by the local queue manager depends on how the remote queue is specified when it is opened.

The following rules are applied:

1. If the remote queue has been defined on the local queue manager through the IBM MQ DEFINE QREMOTE command, the queue that is checked is the name of the remote queue. For example, if a remote queue is defined on queue manager MQS1 as follows:

   ```
   DEFINE QREMOTE(BANK7.CREDIT.REFERENCE)
          RNAME(CREDIT.SCORING.REQUEST)
          RQMNAME(BNK7)
          XMITQ(BANK1.TO.BANK7)
   ```

   In this case, a profile for BANK7.CREDIT.REFERENCE must be defined in the MQQUEUE class.
2. If the *ObjectQMgrName* for the request does not resolve to the local queue manager, a security check is carried out against the resolved (remote) queue manager name except in the case of a cluster queue where the check is made against the cluster queue name.

   For example, the transmission queue BANK1.TO.BANK7 is defined on queue manager MQS1. An MQPUT1 request is then issued on MQS1 specifying *ObjectName* as BANK1.INTERBANK.TRANSFERS and an *ObjectQMgrName* of BANK1.TO.BANK7. In this case, the user performing the request must have access to BANK1.TO.BANK7.
3. If you make an MQPUT request to a queue and specify *ObjectQMgrName* as the name of an alias of the local queue manager, only the queue name is checked for security, not that of the queue manager.

When the message gets to the remote queue manager it might be subject to additional security processing. For more information, see “Segurança para o Sistema de Mensagens Remoto” on page 105.

z/OS *Dead-letter queue security*

Special considerations apply to the dead-letter queue, because many users must be able to put messages on it, but access to retrieve messages must be tightly restricted. You can achieve this by applying different RACF authorities to the dead-letter queue and an alias queue.

Undelivered messages can be put on a special queue called the dead-letter queue. If you have sensitive data that could possibly end up on this queue, you must consider the security implications of this because you do not want unauthorized users to retrieve this data.

Each of the following must be allowed to put messages onto the dead-letter queue:

- Application programs.
- The channel initiator address space and any MCA user IDs. (If the RESLEVEL profile is not present, or is defined so that channel user IDs are checked, the channel user ID also needs authority to put messages on the dead-letter queue.)
- CKTI, the CICS-supplied CICS task initiator.
- CSQQTRMN, the IBM MQ-supplied IMS trigger monitor.

The only application that can retrieve messages from the dead-letter queue should be a 'special' application that processes these messages. However, a problem arises if you give applications RACF UPDATE authority to the dead-letter queue for MQPUT s because they can then automatically retrieve messages from the queue using MQGET calls. You cannot disable the dead-letter queue for get operations because, if you do, not even the 'special' applications could retrieve the messages.

One solution to this problem is set up a two-level access to the dead-letter queue. CKTI, message channel agent transactions or the channel initiator address space, and 'special' applications have direct access; other applications can only access the dead-letter queue through an alias queue. This alias is defined to allow applications to put messages on the dead-letter queue, but not to get messages from it.

This is how it might work:

1. Define the real dead-letter queue with attributes PUT(ENABLED) and GET(ENABLED), as shown in the sample thlqual.SCSQPROC(CSQ4INYG).
2. Give RACF UPDATE authority for the dead-letter queue to the following user IDs:
   - User IDs that the CKTI and the MCAs or channel initiator address space run under.
   - The user IDs associated with the 'special' dead-letter queue processing application.
3. Define an alias queue that resolves to the real dead-letter queue, but give the alias queue these attributes: PUT(ENABLED) and GET(DISABLED). Give the alias queue a name with the same stem as the dead-letter queue name but append the characters ".PUT" to this stem. For example, if the dead-letter queue name is hlq.DEAD.QUEUE, the alias queue name would be hlq.DEAD.QUEUE.PUT.
4. To put a message on the dead-letter queue, an application uses the alias queue. This is what your application must do:
   - Retrieve the name of the real dead-letter queue. To do this, it opens the queue manager object using MQOPEN and then issues an MQINQ to get the dead-letter queue name.
   - Build the name of the alias queue by appending the characters '.PUT' to this name, in this case, hlq.DEAD.QUEUE.PUT.
   - Open the alias queue, hlq.DEAD.QUEUE.PUT.
   - Put the message on the real dead-letter queue by issuing an MQPUT against the alias queue.
5. Give the user ID associated with the application RACF UPDATE authority to the alias, but no access (authority NONE) to the real dead-letter queue. This means that:
   - The application can put messages onto the dead-letter queue using the alias queue.
   - The application cannot get messages from the dead-letter queue using the alias queue because the alias queue is disabled for get operations.

   The application cannot get any messages from the real dead-letter queue either because it does have the correct RACF authority.

Table 34 on page 213 summarizes the RACF authority required for the various participants in this solution.

Table 34. RACF authority to the dead-letter queue and its alias

Associated user IDs	Real dead-letter queue (hlq.DEAD.QUEUE)	Alias dead-letter queue (hlq.DEAD.QUEUE.PUT)
MCA or channel initiator address space and CKTI	UPDATE	NONE
'Special' application (for dead-letter queue processing)	UPDATE	NONE
User-written application user IDs	NONE	UPDATE

If you use this method, the application cannot determine the maximum message length (MAXMSGL) of the dead-letter queue. This is because the MAXMSGL attribute cannot be retrieved from an alias queue. Therefore, your application should assume that the maximum message length is 100 MB, the maximum size IBM MQ for z/OS supports. The real dead-letter queue should also be defined with a MAXMSGL attribute of 100 MB.

Note: User-written application programs do not normally use alternate user authority to put messages on the dead-letter queue. This reduces the number of user IDs that have access to the dead-letter queue.

z/OS *System queue security*

You must set up RACF access to allow certain user IDs access to particular system queues.

Many of the system queues are accessed by the ancillary parts of IBM MQ:

- The CSQUTIL utility
- The message security policy utility (CSQ0UTIL)
- The operations and control panels
- The channel initiator address space (including the Queued Pub/Sub Daemon)
- The mqweb server, used by the IBM MQ Console and REST API.

The user IDs under which these run must be given RACF access to these queues, as shown in Table 35 on page 213.

Table 35. Access required to the SYSTEM queues by IBM MQ

SYSTEM queue	CSQUTIL	CSQ0UTIL	mqweb server	Operations and control panels	Channel initiator for distributed queuing
SYSTEM.ADMIN.CHANNEL.EVENT	-	-	-	-	UPDATE
SYSTEM.ADMIN.COMMAND.QUEUE	-	-	UPDATE	-	-
SYSTEM.BROKER.ADMIN.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.CONTROL.QUEUE	-	-	-	-	ALTER
SYSTEM.BROKER.DEFAULT.STREAM	-	-	-	-	ALTER
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	-	-	-	-	UPDATE
SYSTEM.CHANNEL.INITQ	-	-	-	-	UPDATE
SYSTEM.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.CLUSTER.COMMAND.QUEUE	-	-	-	-	ALTER

Table 35. Access required to the SYSTEM queues by IBM MQ (continued)					
SYSTEM queue	**CSQUTIL**	**CSQ0UTIL**	**mqweb server**	**Operations and control panels**	**Channel initiator for distributed queuing**
SYSTEM.CLUSTER.REPOSITORY.QUEUE	-	-	-	-	UPDATE
SYSTEM.CLUSTER.TRANSMIT.QUEUE	-	-	-	-	ALTER
SYSTEM.COMMAND.INPUT	UPDATE	-	-	UPDATE	UPDATE
SYSTEM.COMMAND.REPLY.*	-	-	-	-	UPDATE
SYSTEM.COMMAND.REPLY.MODEL	UPDATE	-	-	UPDATE	UPDATE
SYSTEM.CSQOREXX.*	-	-	-	UPDATE	-
SYSTEM.CSQUTIL.*	UPDATE	-	-	-	-
SYSTEM.CSQXCMD.*	-	-	-	-	UPDATE
SYSTEM.HIERARCHY.STATE	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.CONTROL	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.PUBS	-	-	-	-	UPDATE
SYSTEM.INTER.QMGR.FANREQ	-	-	-	-	UPDATE
SYSTEM.PROTECTION.ERROR.QUEUE	-	-	-	-	UPDATE
SYSTEM.PROTECTION.POLICY.QUEUE	-	UPDATE “1” on page 214	-	-	READ
SYSTEM.QSG.CHANNEL.SYNCQ	-	-	-	-	UPDATE
SYSTEM.QSG.TRANSMIT.QUEUE	-	-	-	-	UPDATE
SYSTEM.REST.REPLY.QUEUE	-	-	UPDATE	-	-
SYSTEM.BLUEMIX.REGISTRATION.QUEUE	-	-	-	-	UPDATE

Notes:

1. The Advanced Message Security address space user also requires READ access to this queue.

z/OS *API-resource security access quick reference*

A summary of the **MQOPEN**, **MQPUT1**, **MQSUB**, and **MQCLOSE** options and the access required by the different resource security types.

Table 36. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required. Callouts shown like this **(1)** refer to the notes following this table.				
	Minimum RACF access level required			
RACF class:	**MXTOPIC**	**MQQUEUE or MXQUEUE (1)**	**MQADMIN or MXADMIN**	**MQADMIN or MXADMIN**
RACF profile:	**(15 or 16)**	**(2)**	**(3)**	**(4)**
MQOPEN option				

Table 36. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required. Callouts shown like this **(1)** refer to the notes following this table. *(continued)*

	Minimum RACF access level required			
RACF class:	**MXTOPIC**	**MQQUEUE or MXQUEUE (1)**	**MQADMIN or MXADMIN**	**MQADMIN or MXADMIN**
RACF profile:	**(15 or 16)**	**(2)**	**(3)**	**(4)**
MQOO_INQUIRE		READ **(5)**	No check	No check
MQOO_BROWSE		READ	No check	No check
MQOO_INPUT_*		UPDATE	No check	No check
MQOO_SAVE_ALL_CONTEXT **(6)**		UPDATE	No check	No check
MQOO_OUTPUT (USAGE=NORMAL) **(7)**		UPDATE	No check	No check
MQOO_PASS_IDENTITY_CONTEXT **(8)**		UPDATE	READ	No check
MQOO_PASS_ALL_CONTEXT **(8) (9)**		UPDATE	READ	No check
MQOO_SET_IDENTITY_CONTEXT **(8) (9)**		UPDATE	UPDATE	No check
MQOO_SET_ALL_CONTEXT **(8) (10)**		UPDATE	CONTROL	No check
MQOO_OUTPUT (USAGE (XMITQ) **(11)**		UPDATE	CONTROL	No check
MQOO_OUTPUT (topic object)	UPDATE **(16)**			
MQOO_OUTPUT (alias queue to topic object)	UPDATE **(16)**	UPDATE		
MQOO_SET		ALTER	No check	No check
MQOO_ALTERNATE_USER_AUTHORITY		**(12)**	**(12)**	UPDATE
MQPUT1 option				
Put on a normal queue **(7)**		UPDATE	No check	No check
MQPMO_PASS_IDENTITY_CONTEXT		UPDATE	READ	No check
MQPMO_PASS_ALL_CONTEXT		UPDATE	READ	No check
MQPMO_SET_IDENTITY_CONTEXT		UPDATE	UPDATE	No check
MQPMO_SET_ALL_CONTEXT		UPDATE	CONTROL	No check
MQOO_OUTPUT Put on a transmission queue **(11)**		UPDATE	CONTROL	No check
MQOO_OUTPUT (topic object)	UPDATE **(16)**			
MQOO_OUTPUT (alias queue to topic object)	UPDATE **(16)**	UPDATE		
MQPMO_ALTERNATE_USER_AUTHORITY		**(13)**	**(13)**	UPDATE
MQCLOSE option				
MQCO_DELETE **(14)**		ALTER	No check	No check
MQCO_DELETE_PURGE **(14)**		ALTER	No check	No check
MQCO_REMOVE_SUB	ALTER **(15)**			
MQSUB option				

Table 36. MQOPEN, MQPUT1, MQSUB, and MQCLOSE options and the security authorization required. Callouts shown like this **(1)** refer to the notes following this table. *(continued)*

	Minimum RACF access level required			
RACF class:	**MXTOPIC**	**MQQUEUE or MXQUEUE (1)**	**MQADMIN or MXADMIN**	**MQADMIN or MXADMIN**
RACF profile:	**(15 or 16)**	**(2)**	**(3)**	**(4)**
MQSO_CREATE	ALTER **(15)**	**(17)**	**(18)**	
MQSO_ALTER	ALTER **(15)**	**(17)**	**(18)**	
MQSO_RESUME	READ **(15)**	**(17)**	No check	
MQSO_ALTERNATE_USER_AUTHORITY				UPDATE
MQSO_SET_IDENTITY_CONTEXT			**(18)**	

Note:

1. This option is not restricted to queues. Use the MQNLIST or MXNLIST class for namelists, and the MQPROC or MXPROC class for processes.
2. Use RACF profile: hlq.resourcename
3. Use RACF profile: hlq.CONTEXT.queuename
4. Use RACF profile: hlq.ALTERNATE.USER. `alternateuserid`

   `alternateuserid` is the user identifier that is specified in the *`AlternateUserId`* field of the object descriptor. Note that up to 12 characters of the *`AlternateUserId`* field are used for this check, unlike other checks where only the first 8 characters of a user identifier are used.
5. No check is made when opening the queue manager for inquiries.
6. MQOO_INPUT_* must be specified as well. This is valid for a local, model or alias queue.
7. This check is done for a local or model queue that has a **Usage** queue attribute of MQUS_NORMAL, and also for an alias or remote queue (that is defined to the connected queue manager.) If the queue is a remote queue that is opened specifying an *`ObjectQMgrName`* (not the name of the connected queue manager) explicitly, the check is carried out against the queue with the same name as *`ObjectQMgrName`* (which must be a local queue with a **Usage** queue attribute of MQUS_TRANSMISSION).
8. MQOO_OUTPUT must be specified as well.
9. MQOO_PASS_IDENTITY_CONTEXT is implied as well by this option.
10. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT and MQOO_SET_IDENTITY_CONTEXT are implied as well by this option.
11. This check is done for a local or model queue that has a **Usage** queue attribute of MQUS_TRANSMISSION, and is being opened directly for output. It does not apply if a remote queue is being opened.
12. At least one of MQOO_INQUIRE, MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT or MQOO_SET must be specified as well. The check carried out is the same as that for the other options specified.
13. The check carried out is the same as that for the other options specified.
14. This applies only for permanent dynamic queues that have been opened directly, that is, not opened through a model queue. No security is required to delete a temporary dynamic queue.
15. Use RACF profile hlq.SUBSCRIBE.topicname.
16. Use RACF profile hlq.PUBLISH.topicname.
17. If on the MQSUB request you specified a destination queue for the publications to be sent to, then a security check is carried out against that queue to ensure that you have put authority to that queue.

18. If on the MQSUB request, with MQSO_CREATE or MQSO_ALTER options specified, you want to set any of the identity context fields in the MQSD structure, you also need to specify the MQSO_SET_IDENTITY_CONTEXT option and you also need the appropriate authority to the context profile for the destination queue.

z/OS *Profiles for topic security*

If topic security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

The concept of topic security within a topic tree is described in Publish/subscribe security.

If topic security is active, you must perform the following actions:

- Define profiles in the **MXTOPIC** or **GMXTOPIC** classes.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use topics.

Profiles for topic security take the form:

```
hlq.SUBSCRIBE.topicname
hlq.PUBLISH.topicname
```

where

- `hlq` is either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name).
- `topicname` is the name of the topic administration node in the topic tree, associated either with the topic being subscribed to through an MQSUB call, or being published to through an MQOPEN call.

A profile prefixed by the queue manager name controls access to a single topic on that queue manager. A profile prefixed by the queue sharing group name controls access to one or more topics with that topic name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that topic on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

Subscribe

To subscribe to a topic, you need access to both the topic you are trying to subscribe to, and the destination queue for the publications.

When you issue an MQSUB request, the following security checks take place:

- Whether you have the appropriate level of access to subscribe to that topic, and also that the destination queue (if specified) is opened for output
- Whether you have the appropriate level of access to that destination queue.

Table 37. Access level required for topic security to subscribe

MQSUB option	**RACF access required to `hlq.SUBSCRIBE.topicname` profile in MXTOPIC class**
MQSO_CREATE and MQSO_ALTER	ALTER
MQSO_RESUME	READ

Table 38. Additional authority required to subscribe using a non-managed destination queue	
MQSUB option	**RACF access required to `hlq.CONTEXT.queuename` profile in MQADMIN or MXADMIN class**
MQSO_CREATE, MQSO_ALTER, and MQSO_RESUME	UPDATE
	RACF access required to `hlq.queuename` profile in MQQUEUE or MXQUEUE class
MQSO_CREATE and MQSO_ALTER	UPDATE
	RACF access required to `hlq.ALTERNATE.USER.alternateuserid` profile in MQADMIN or MXADMIN class
MQSO_ALTERNATE_USER_AUTHORITY	UPDATE

Considerations for managed queues for subscriptions

A security check is carried out to see if you are allowed to subscribe to the topic. However, no security checks are carried out when the managed queue is created, or to determine if you have access to put messages to this destination queue.

You cannot close delete a managed queue.

The model queues used are: `SYSTEM.DURABLE.MODEL.QUEUE` and `SYSTEM.NDURABLE.MODEL.QUEUE`.

The managed queues created from these model queues
are of the form `SYSTEM.MANAGED.DURABLE.A346EF00367849A0` and
`SYSTEM.MANAGED.NDURABLE.A346EF0036785EA0` where the last qualifier is unpredictable.

Do not give any user access to these queues. The queues can be protected using generic profiles of the form `SYSTEM.MANAGED.DURABLE.*` and `SYSTEM.MANAGED.NDURABLE.*` with no authorities granted.

Messages can be retrieved from these queues using the handle returned on the MQSUB request.

If you explicitly issue an MQCLOSE call for a subscription with the `MQCO_REMOVE_SUB` option specified, and you did not create the subscription you are closing under this handle, a security check is performed at the time of closure to ensure that you have the correct authority to perform the operation.

Table 39. Access level required to profiles for topic security for closure of a subscribe operation	
MQCLOSE option	**RACF access required to `hlq.SUBSCRIBE.topicname` profile in MXTOPIC class**
MQCO_REMOVE_SUB	ALTER

Publish

To publish on a topic you need access to the topic and, if you are using alias queues, to the alias queue as well.

Table 40. Access level required for topic security to publish	
MQOPEN or MQPUT1 option	**RACF access required to `hlq.PUBLISH.topicname` profile in MXTOPIC class**
MQOO_OUTPUT or MQPUT1	UPDATE

Table 41. Access level required to open an alias queue that resolves to a topic	
MQOPEN or MQPUT1 option	**RACF access required to `hlq.queuename` profile in MQQUEUE or MXQUEUE class for the alias queue**
MQOO_OUTPUT or MQPUT1	UPDATE

For details of how topic security operates when an alias queue that resolves to a topic name is opened for publish, see “Considerations for alias queues that resolve to topics for a publish operation” on page 219.

When you consider alias queues used for destination queues for PUT or GET restrictions, see “Considerations for alias queues” on page 209.

If the RACF access level that an application has to a topic security profile is changed, the changes take effect only for any new object handles obtained (that is, a new MQSUB or MQOPEN) for that topic. Those handles already in existence at the time of the change retain their existing access to the topic. Also, existing subscribers retain their access to any subscriptions that they have already made.

Considerations for alias queues that resolve to topics for a publish operation

When you issue an MQOPEN or MQPUT1 call for an alias queue that resolves to a topic, IBM MQ makes two resource checks:

- The first one against the alias queue name specified in the object descriptor (MQOD) on the MQOPEN or MQPUT1 call.
- The second against the topic to which the alias queue resolves

You must be aware that this behavior is different from the behavior you get when alias queues resolve to other queues. You need the correct access to both profiles in order for the publish action to proceed.

System topic security

The following system topics are accessed by the channel initiator address space.

The user IDs under which this runs must be given RACF access to these queues, as shown in Table 42 on page 219.

Table 42. Access required to the SYSTEM topics		
SYSTEM topic	**Profile**	**Channel initiator for distributed queuing**
SYSTEM.BROKER.ADMIN.STREAM	hlq.PUBLISH.topicname	UPDATE
SYSTEM.BROKER.ADMIN.STREAM	hlq.SUBSCRIBE.topicname	ALTER

z/OS *Profiles for processes*

If process security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

If process security is active, you must:

- Define profiles in the **MQPROC** or **GMQPROC** classes if using uppercase profiles.
- Define profiles in the **MXPROC** or **GMXPROC** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles, so that they can issue IBM MQ API requests that use processes.

Profiles for processes take the form:

```
hlq.processname
```

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue sharing group name), and processname is the name of the process being opened.

A profile prefixed by the queue manager name controls access to a single process definition on that queue manager. A profile prefixed by the queue sharing group name controls access to one or more process definitions with that name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that process definition on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access required for opening a process.

Table 43. Access levels for process security

MQOPEN option	**RACF access level required to hlq.processname**
MQOO_INQUIRE	READ

For example, on queue manager MQS9, the RACF group INQVPRC must be able to inquire (MQINQ) on all processes starting with the letter V. The RACF definitions for this would be:

```
RDEFINE MQPROC MQS9.V* UACC(NONE)
PERMIT MQS9.V* CLASS(MQPROC) ID(INQVPRC) ACCESS(READ)
```

Alternate user security might also be active, depending on the open options specified when a process definition object is opened.

z/OS *Profiles for namelists*

If namelist security is active, you define profiles in the appropriate classes and give the necessary groups or user IDs access to these profiles.

If namelist security is active, you must:

- Define profiles in the **MQNLIST** or **GMQNLIST** classes if using uppercase profiles.
- Define profiles in the **MXNLIST** or **GMXNLIST** classes if using mixed case profiles.
- Permit the necessary groups or user IDs access to these profiles.

Profiles for namelists take the form:

```
hlq.namelistname
```

where hlq can be either qmgr-name (queue manager name) or qsg-name (queue sharing group name), and namelistname is the name of the namelist being opened.

A profile prefixed by the queue manager name controls access to a single namelist on that queue manager. A profile prefixed by the queue sharing group name controls access to access to one or more namelists with that name on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that namelist on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access required for opening a namelist.

Table 44. Access levels for namelist security

MQOPEN option	RACF access level required to hlq.namelistname
MQOO_INQUIRE	READ

For example, on queue manager (or queue sharing group) PQM3, the RACF group DEPT571 must be able to inquire (`MQINQ`) on these namelists:

- All namelists starting with "DEPT571".
- PRINTER/DESTINATIONS/DEPT571
- AGENCY/REQUEST/QUEUES
- WAREHOUSE.BROADCAST

The RACF definitions to do this are:

```
RDEFINE MQNLIST PQM3.DEPT571.** UACC(NONE)
PERMIT PQM3.DEPT571.** CLASS(MQNLIST) ID(DEPT571) ACCESS(READ)

RDEFINE GMQNLIST NLISTS.FOR.DEPT571 UACC(NONE)
        ADDMEM(PQM3.PRINTER/DESTINATIONS/DEPT571,
               PQM3.AGENCY/REQUEST/QUEUES,
               PQM3.WAREHOUSE.BROADCAST)
PERMIT NLISTS.FOR.DEPT571 CLASS(GMQNLIST) ID(DEPT571) ACCESS(READ)
```

Alternate user security might be active, depending on the options specified when a namelist object is opened.

System namelist security

Many of the system namelists are accessed by the ancillary parts of IBM MQ:

- The CSQUTIL utility
- The operations and control panels
- The channel initiator address space (including the Queued Publish/Subscribe Daemon)

The user IDs under which these run must be given RACF access to these namelists, as shown in Table 45 on page 221.

Table 45. Access required to the SYSTEM namelists by IBM MQ

SYSTEM namelist	CSQUTIL	Operations and control panels	Channel initiator for distributed queuing
SYSTEM.QPUBSUB.QUEUE.NAMELIST	-	-	READ
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	-	-	READ

z/OS *Profiles for alternate user security*

If alternate user security is active, you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles.

For more information about *AlternateUserId*, see AlternateUserID (MQCHAR12).

If alternate user security is active, you must:

- Define profiles in the MQADMIN or GMQADMIN classes if you are using uppercase profiles.
- Define profiles in the MXADMIN or GMXADMIN classes if you are using mixed case profiles.

Permit the necessary groups or user IDs access to these profiles, so that they can use the ALTERNATE_USER_AUTHORITY options when the object is opened.

Profiles for alternate user security can be specified at subsystem level or at queue sharing group level and take the following form:

```
hlq.ALTERNATE.USER.alternateuserid
```

Where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), and `alternateuserid` is the value of the *`AlternateUserId`* field in the object descriptor.

A profile prefixed by the queue manager name controls use of an alternative user ID on that queue manager. A profile prefixed by the queue sharing group name controls use of an alternative user ID on all queue managers within the queue sharing group. This alternative user ID can be used on any queue manager within the queue sharing group by a user that has the correct access. This access can be overridden on an individual queue manager by defining a queue manager level profile for that alternative user ID on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

The following table shows the access when specifying an alternative user option.

Table 46. Access levels for alternate user security

MQOPEN, MQSUB, or MQPUT1 option	**RACF access level required**
MQOO_ALTERNATE_USER_AUTHORITY MQSO_ALTERNATE_USER_AUTHORITY MQPMO_ALTERNATE_USER_AUTHORITY	UPDATE

In addition to alternate user security checks, other security checks for queue, process, namelist, and context security can also be made. The alternative user ID, if provided, is only used for security checks on queue, process definition, or namelist resources. For alternate user and context security checks, the user ID requesting that the check is used. For details about how user IDs are handled, see “User IDs for security checking on z/OS” on page 243. For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 36 on page 214.

An alternative user profile gives the requesting user ID access to resources associated with the user ID specified in the alternative user ID. For example, the payroll server running under user ID PAYSERV on queue manager QMPY processes requests from personnel user IDs, all of which start with PS. To cause the work performed by the payroll server to be carried out under the user ID of the requesting user, alternative user authority is used. The payroll server knows which user ID to specify as the alternative user ID because the requesting programs generate messages using the MQPMO_DEFAULT_CONTEXT put message option. See “User IDs for security checking on z/OS” on page 243 for more details about from where alternative user IDs are obtained.

The following example RACF definitions enable the server program to specify alternative user IDs starting with the characters PS:

```
RDEFINE MQADMIN QMPY.ALTERNATE.USER.PS* UACC(NONE)
PERMIT QMPY.ALTERNATE.USER.PS* CLASS(MQADMIN) ID(PAYSERV) ACCESS(UPDATE)
```

Note:

1. The *AlternateUserId* fields in the object descriptor and subscription descriptor are 12 bytes long. All 12 bytes are used in the profile checks, but only the first 8 bytes are used as the user ID by IBM MQ. If this user ID truncation is not desirable, application programs making the request must translate any alternative user ID over 8 bytes into something more appropriate.
2. If you specify MQOO_ALTERNATE_USER_AUTHORITY, MQSO_ALTERNATE_USER_AUTHORITY, or MQPMO_ALTERNATE_USER_AUTHORITY and you do not specify an *AlternateUserId* field in the object descriptor, a user ID of blanks is used. For the purposes of the alternate user security check the user ID used for the *AlternateUserId* qualifier is -BLANK-. For example `RDEF MQADMIN hlq.ALTERNATE.USER.-BLANK-`.

   If the user is allowed to access this profile, all further checks are made with a user ID of blanks. For details of blank user IDs, see “Blank user IDs and UACC levels” on page 251.

The administration of alternative user IDs is easier if you have a naming convention for user IDs that enables you to use generic alternative user profiles. If they do not, you can use the RACF RACVAR feature. For details about using RACVAR, see the z/OS Security Server RACF documentation..

When a message is put to a queue that has been opened with alternative user authority and the context of the message has been generated by the queue manager, the MQMD_USER_IDENTIFIER field is set to the alternative user ID.

z/OS *Profiles for context security*

If context security is active, to control access to the message context information you must define profiles in the appropriate classes and permit the necessary groups or user IDs access to those profiles. The message context is contained within the message descriptor (MQMD).

Using profiles for context security

If context security is active, to permit users to access context information for messages on a particular queue, or when publishing to a particular topic, you must define a profile in one of the following classes:

- The `MQADMIN` class if using uppercase profiles.
- The `MXADMIN` class if using mixed-case profiles.

Profiles for context security can be specified at subsystem level or at queue sharing group level and take the following form:

```
hlq.CONTEXT.queuename
hlq.CONTEXT.topicname
```

where *hlq* can be either the queue manager name or the queue sharing group name, and *queuename* and *topicname* can be either the full or generic name of the queue or topic you want to define the context profile for.

A profile prefixed by the queue manager name, and with ** specified as the queue or topic name, allows control for context security on all queues and topics belonging to that queue manager. This can be overridden on an individual queue or topic by defining a specific profile for context on that queue or topic.

A profile prefixed by the queue sharing group name, and with ** specified as the queue or topic name, allows control for context on all queues and topics belonging to the queue managers within the queue sharing group. This can be overridden on an individual queue manager by defining a queue manager level profile for context on that queue manager, by specifying a profile prefixed by the queue manager name. It can also be overridden on an individual queue or topic by specifying a profile suffixed with the queue or topic name.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

You must permit the necessary groups or user IDs access to this profile. The following table shows the access level required, depending on the specification of the context options when the queue is opened.

Table 47. Access levels for context security	
MQOPEN or MQPUT1 option	**RACF access level required to hlq.CONTEXT.queuename or hlq.CONTEXT.topicname**
MQPMO_NO_CONTEXT	No context security check
MQPMO_DEFAULT_CONTEXT	No context security check
MQOO_SAVE_ALL_CONTEXT	No context security check
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	READ
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	READ
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	UPDATE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	CONTROL
MQOO_OUTPUT or MQPUT1 (USAGE(XMITQ))	CONTROL
MQSUB option	
MQSO_SET_IDENTITY_CONTEXT **(Note 2)**	UPDATE

Note:

1. The user IDs used for distributed queuing require `CONTROL` access to `hlq.CONTEXT.queuename` to put messages on the destination queue. See "User IDs used by the channel initiator" on page 246 for information about the user IDs used.
2. If on the MQSUB request, with `MQSO_CREATE` or `MQSO_ALTER` options specified, you want to set any of the identity context fields in the MQSD structure, you need to specify the `MQSO_SET_IDENTITY_CONTEXT` option. You require also, the appropriate authority to the context profile for the destination queue.

If you put commands on the system-command input queue, use the default context put message option to associate the correct user ID with the command.

For example, the IBM MQ-supplied utility program CSQUTIL can be used to offload and reload messages in queues. When offloaded messages are restored to a queue, the CSQUTIL utility uses the `MQOO_SET_ALL_CONTEXT` option to return the messages to their original state. In addition to the queue security required by this open option, context authority is also required. For example, if this authority is required by the group BACKGRP on queue manager MQS1, this would be defined by:

```
RDEFINE MQADMIN MQS1.CONTEXT.** UACC(NONE)
PERMIT MQS1.CONTEXT.** CLASS(MQADMIN) ID(BACKGRP) ACCESS(CONTROL)
```

Depending on the options specified, and the types of security performed, other types of security checks might also occur when the queue is opened. These include queue security (see "Profiles for queue security" on page 207), and alternate user security (see "Profiles for alternate user security" on page 221). For a summary table showing the open options and the security checks required when queue, context and alternate user security are all active, see Table 36 on page 214.

System queue context security

Many of the system queues are accessed by the ancillary parts of IBM MQ, for example the channel initiator address space, and the mqweb server used by the IBM MQ Console and REST API.

The user IDs under which these run under must be given RACF access to these queues, as shown in Table 48 on page 225.

Table 48. Access required to the SYSTEM queues for context operations

SYSTEM queue	Channel initiator for distributed queuing	mqweb server
SYSTEM.ADMIN.COMMAND.QUEUE	-	CONTROL
SYSTEM.BROKER.CONTROL.QUEUE	CONTROL	-
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	CONTROL	-
SYSTEM.CHANNEL.SYNCQ	CONTROL	-
SYSTEM.CLUSTER.COMMAND.QUEUE	CONTROL	-
SYSTEM.CLUSTER.TRANSMIT.QUEUE	CONTROL	-

z/OS *Profiles for command security*

To enable security checking for commands, add profiles to the MQCMDS class. The profile names are based on the MQSC commands but control both MQSC and PCF commands. Profiles can apply to a queue manager or a queue sharing group.

If you want security checking for commands (so you have not defined the command security switch profile hlq.NO.CMD.CHECKS) you must add profiles to the MQCMDS class.

The same security profiles control both MQSC and PCF commands. The names of the RACF profiles for command security checking are based on the MQSC command names themselves. These profiles take the form:

```
hlq.verb.pkw
```

Where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name), `verb` is the verb part of the command name, for example ALTER, and `pkw` is the object type, for example QLOCAL for a local queue.

Thus, the profile name for the ALTER QLOCAL command in subsystem CSQ1 is:

```
CSQ1.ALTER.QLOCAL
```

You can use generic profiles to protect sets of commands so that you have fewer profiles to maintain and, therefore, fewer access lists. Consider creating a generic profile that applies to all commands not protected by a more specific profile. Define this profile with UACC(NONE) and grant ALTER access only to the RACF groups containing administrators. You might then create a generic profile applicable to all DISPLAY commands and grant widespread access to it. Between these extremes, you might identify groups of users needing access to certain sets of commands, in which case you can create profiles for those sets and grant access to RACF groups representing those classes of user. Avoid giving users access to commands they do not require: Apply the principle of least privilege, so that users only have access to the commands that are required for their jobs.

A profile prefixed by the queue manager name controls the use of the command on that queue manager. A profile prefixed by the queue sharing group name controls the use of the command on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that command on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

By setting up command profiles at queue manager level, a user can be restricted from issuing commands on a particular queue manager. Alternatively, you can define one profile for a queue sharing group for each command verb, and all security checks take place against that profile instead of individual queue managers.

If both subsystem security and queue sharing group security are active and a local profile is not found, a command security check is performed to see if the user has access to a queue sharing group profile.

If you use the CMDSCOPE attribute to route a command to other queue managers in a queue sharing group, security is checked on each queue manager where the command is run, but not necessarily on the queue manager where the command is entered.

Table 49 on page 226 shows, for each IBM MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

Table 50 on page 231 shows, for each IBM MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

Table 49. MQSC commands, profiles, and their access levels

Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
ALTER AUTHINFO	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
ALTER BUFFPOOL	hlq.ALTER.BUFFPOOL	ALTER	No check	-
ALTER CFSTRUCT	hlq.ALTER.CFSTRUCT	ALTER	No check	-
ALTER CHANNEL	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
ALTER NAMELIST	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
ALTER PROCESS	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
ALTER PSID	hlq.ALTER.PSID	ALTER	No check	-
ALTER QALIAS	hlq.ALTER.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
ALTER QLOCAL“5” on page 231	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	No check	-
ALTER QMODEL“5” on page 231	hlq.ALTER.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QREMOTE	hlq.ALTER.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
ALTER SECURITY	hlq.ALTER.SECURITY	ALTER	No check	-
ALTER SMDS	hlq.ALTER.SMDS	ALTER	No check	-
ALTER STGCLASS	hlq.ALTER.STGCLASS	ALTER	No check	-
ALTER SUB	hlq.ALTER.SUB	ALTER	No check	-
ALTER TOPIC	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
ALTER TRACE	hlq.ALTER.TRACE	ALTER	No check	-
ARCHIVE LOG	hlq.ARCHIVE.LOG	CONTROL	No check	-

Table 49. MQSC commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
BACKUP CFSTRUCT	hlq.BACKUP.CFSTRUCT	CONTROL	No check	-
CLEAR QLOCAL	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
CLEAR TOPICSTR “3” on page 230	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
DEFINE AUTHINFO	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DEFINE BUFFPOOL	hlq.DEFINE.BUFFPOOL	ALTER	No check	-
DEFINE CFSTRUCT	hlq.DEFINE.CFSTRUCT	ALTER	No check	-
DEFINE CHANNEL	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DEFINE LOG	hlq.DEFINE.LOG	ALTER	No check	-
DEFINE MAXSMSGS	hlq.DEFINE.MAXSMSGS	ALTER	No check	-
DEFINE NAMELIST	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
DEFINE PROCESS	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DEFINE PSID	hlq.DEFINE.PSID	ALTER	No check	-
DEFINE QALIAS	hlq.DEFINE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QLOCAL“5” on page 231	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QMODEL“5” on page 231	hlq.DEFINE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QREMOTE	hlq.DEFINE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DEFINE STGCLASS	hlq.DEFINE.STGCLASS	ALTER	No check	-
DEFINE SUB	hlq.DEFINE.SUB	ALTER	No check	-
DEFINE TOPIC	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DELETE AUTHINFO	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
DELETE BUFFPOOL	hlq.DELETE.BUFFPOOL	ALTER	No check	-
DELETE CFSTRUCT	hlq.DELETE.CFSTRUCT	ALTER	No check	-
DELETE CHANNEL	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
DELETE NAMELIST	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
DELETE PROCESS	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
DELETE PSID	hlq.DELETE.PSID	ALTER	No check	-
DELETE QALIAS	hlq.DELETE.QALIAS	ALTER	hlq.QUEUE.queue	ALTER
DELETE QLOCAL	hlq.DELETE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER

Table 49. MQSC commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
DELETE QMODEL	hlq.DELETE.QMODEL	ALTER	hlq.QUEUE.queue	ALTER
DELETE QREMOTE	hlq.DELETE.QREMOTE	ALTER	hlq.QUEUE.queue	ALTER
DELETE STGCLASS	hlq.DELETE.STGCLASS	ALTER	No check	-
DELETE SUB	hlq.DELETE.SUB	ALTER	No check	-
DELETE TOPIC	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
DISPLAY ARCHIVE “1” on page 230	hlq.DISPLAY.ARCHIVE	READ	No check	-
DISPLAY AUTHINFO	hlq.DISPLAY.AUTHINFO	READ	No check	-
DISPLAY CFSTATUS	hlq.DISPLAY.CFSTATUS	READ	No check	-
DISPLAY CFSTRUCT	hlq.DISPLAY.CFSTRUCT	READ	No check	-
DISPLAY CHANNEL	hlq.DISPLAY.CHANNEL	READ	No check	-
DISPLAY CHINIT	hlq.DISPLAY.CHINIT	READ	No check	-
DISPLAY CHLAUTH	hlq.DISPLAY.CHLAUTH	READ	No check	-
DISPLAY CHSTATUS	hlq.DISPLAY.CHSTATUS	READ	No check	-
DISPLAY CLUSQMGR	hlq.DISPLAY.CLUSQMGR	READ	No check	-
DISPLAY CMDSERV	hlq.DISPLAY.CMDSERV	READ	No check	-
DISPLAY CONN “1” on page 230	hlq.DISPLAY.CONN	READ	No check	-
DISPLAY GROUP	hlq.DISPLAY.GROUP	READ	No check	-
DISPLAY LOG “1” on page 230	hlq.DISPLAY.LOG	READ	No check	-
DISPLAY MAXSMSGS	hlq.DISPLAY.MAXSMSGS	READ	No check	-
DISPLAY NAMELIST	hlq.DISPLAY.NAMELIST	READ	No check	-
DISPLAY PROCESS	hlq.DISPLAY.PROCESS	READ	No check	-
DISPLAY PUBSUB	hlq.DISPLAY.PUBSUB	READ	No check	-
DISPLAY QALIAS	hlq.DISPLAY.QALIAS	READ	No check	-
DISPLAY QCLUSTER	hlq.DISPLAY.QCLUSTER	READ	No check	-
DISPLAY QLOCAL	hlq.DISPLAY.QLOCAL	READ	No check	-
DISPLAY QMGR	hlq.DISPLAY.QMGR	READ	No check	-
DISPLAY QMODEL	hlq.DISPLAY.QMODEL	READ	No check	-
DISPLAY QREMOTE	hlq.DISPLAY.QREMOTE	READ	No check	-
DISPLAY QSTATUS	hlq.DISPLAY.QSTATUS	READ	No check	-
DISPLAY QUEUE	hlq.DISPLAY.QUEUE	READ	No check	-

Table 49. MQSC commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
DISPLAY SBSTATUS	hlq.DISPLAY.SBSTATUS	READ	No check	-
DISPLAY SMDS	hlq.DISPLAY.SMDS	READ	No check	-
DISPLAY SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	No check	-
DISPLAY SUB	hlq.DISPLAY.SUB	READ	No check	-
DISPLAY SECURITY	hlq.DISPLAY.SECURITY	READ	No check	-
DISPLAY STGCLASS	hlq.DISPLAY.STGCLASS	READ	No check	-
DISPLAY SYSTEM "1" on page 230	hlq.DISPLAY.SYSTEM	READ	No check	-
DISPLAY THREAD	hlq.DISPLAY.THREAD	READ	No check	-
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	No check	-
DISPLAY TOPIC	hlq.DISPLAY.TOPIC	READ	No check	-
DISPLAY TPSTATUS	hlq.DISPLAY.TPSTATUS	READ	No check	-
DISPLAY TRACE	hlq.DISPLAY.TRACE	READ	No check	-
DISPLAY USAGE "1" on page 230	hlq.DISPLAY.USAGE	READ	No check	-
MOVE QLOCAL	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
PING CHANNEL	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RECOVER BSDS	hlq.RECOVER.BSDS	CONTROL	No check	-
RECOVER CFSTRUCT	hlq.RECOVER.CFSTRUCT	CONTROL	No check	-
REFRESH CLUSTER	hlq.REFRESH.CLUSTER	ALTER	No check	-
REFRESH QMGR	hlq.REFRESH.QMGR	ALTER	No check	-
REFRESH SECURITY	hlq.REFRESH.SECURITY	ALTER	No check	-
RESET CFSTRUCT	hlq.RESET.CFSTRUCT	CONTROL	No check	-
RESET CHANNEL	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RESET CLUSTER	hlq.RESET.CLUSTER	CONTROL	No check	-
RESET QMGR	hlq.RESET.QMGR	CONTROL	No check	-
RESET QSTATS	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
RESET SMDS	hlq.RESET.SMDS	CONTROL	No check	-
RESET TPIPE	hlq.RESET.TPIPE	CONTROL	No check	-
RESOLVE CHANNEL	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
RESOLVE INDOUBT	hlq.RESOLVE.INDOUBT	CONTROL	No check	-
RESUME QMGR	hlq.RESUME.QMGR	CONTROL	No check	-

Table 49. MQSC commands, profiles, and their access levels (continued)				
Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
RVERIFY SECURITY	hlq.RVERIFY.SECURITY	ALTER	No check	-
SET ARCHIVE	hlq.SET.ARCHIVE	CONTROL	No check	-
SET CHLAUTH	hlq.SET.CHLAUTH	CONTROL	No check	-
SET LOG	hlq.SET.LOG	CONTROL	No check	-
SET SYSTEM	hlq.SET.SYSTEM	CONTROL	No check	-
START CHANNEL	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
START CHINIT "4" on page 230	hlq.START.CHINIT	CONTROL	No check	-
START CMDSERV	hlq.START.CMDSERV	CONTROL	No check	-
START LISTENER	hlq.START.LISTENER	CONTROL	No check	-
START QMGR	None "2" on page 230	-	-	-
START SMDSCONN	hlq.START.SMDSCONN	CONTROL	No check	-
START TRACE	hlq.START.TRACE	CONTROL	No check	-
STOP CHANNEL	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
STOP CHINIT	hlq.STOP.CHINIT	CONTROL	No check	-
STOP CMDSERV	hlq.STOP.CMDSERV	CONTROL	No check	-
STOP LISTENER	hlq.STOP.LISTENER	CONTROL	No check	-
STOP QMGR	hlq.STOP.QMGR	CONTROL	No check	-
STOP SMDSCONN	hlq.STOP.SMDSCONN	CONTROL	No check	-
STOP TRACE	hlq.STOP.TRACE	CONTROL	No check	-
SUSPEND QMGR	hlq.SUSPEND.QMGR	CONTROL	No check	-

Notes:

1. These commands might be issued internally by the queue manager; no authority is checked in these cases.
2. IBM MQ does not check the authority of the user who issues the START QMGR command. However, you can use RACF, or your alternative security facilities to control access to the START xxxxMSTR command that is issued as a result of the START QMGR command.

   This is done by controlling access to the MVS.START.STC.xxxxMSTR profile in the RACF operator commands (OPERCMDS) class. For details of this procedure, see Granting the user access to the RACF OPERCMDS class in *z/OS MVS Planning: Operations*. If you use this technique, and an unauthorized user tries to start the queue manager, it terminates with a reason code of 00F30216.
3. The **hlq.TOPIC.topic** resource refers to the Topic object derived from the TOPICSTR. For more details, see "Segurança de Publicação/Assinatura" on page 492
4. In IBM MQ for z/OS, the resource name MVS.START.STC.CSQ1CHIN has an additional JOBNAME qualifier appended. This can cause problems when starting the channel initiator.

To resolve the problem replace MVS.START.STC. *ssid* CHIN with a profile for a resource named MVS.START.STC. *ssid* CHIN **.*** or MVS.START.STC. *ssid* CHIN. *ssid* CHIN where *ssid* is the subsystem ID for the queue manager. This requires RACF UPDATE authority. For more details, see MVS™ Commands, RACF Access Authorities, and Resource Names in *z/OS MVS Planning: Operations*.

The START for *ssid* MSTR does not include the JOBNAME= parameter. For consistency, you might want to update the profile for MVS.START.STC.ssidMSTR to MVS.START.STC.ssidMSTR.*.

5. Setting the queue attribute STREAMQ to a non blank value also requires ALTER access level to MQADMIN or MXADMIN for `hlq.ALTER.streamQ`.

Table 50. PCF commands, profiles, and their access levels

Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
Backup CF Structure	hlq.BACKUP.CFSTRUCT	CONTROL	No check	-
Change Authentication Information Object	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Change CF Structure	hlq.ALTER.CFSTRUCT	ALTER	No check	-
Change Channel	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Change Namelist	hlq.ALTER.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Change Process	hlq.ALTER.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Change Queue“2” on page 234	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Change Queue Manager	hlq.ALTER.QMGR	ALTER	No check	-
Change Security	hlq.ALTER.SECURITY	ALTER	No check	-
Change SMDS	hlq.ALTER.SMDS	ALTER	No check	-
Change Storage Class	hlq.ALTER.STGCLASS	ALTER	No check	-
Change Subscription	hlq.ALTER.SUB	ALTER	No check	-
Change Topic	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Clear Queue	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Clear Topic String “1” on page 234	hlq.CLEAR.TOPICSTR	ALTER	hlq.TOPIC.topic	ALTER
Copy Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Copy CF Structure	hlq.DEFINE.CFSTRUCT	ALTER	No check	-
Copy Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Copy Namelist	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Copy Process	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Copy Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Copy Subscription	hlq.DEFINE.SUB	ALTER	No check	-
Copy Storage Class	hlq.DEFINE.STGCLASS	ALTER	No check	-
Copy Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Create Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER

Table 50. PCF commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Create CF Structure	hlq.DEFINE.CFSTRUCT	ALTER	No check	-
Create Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Create Namelist	hlq.DEFINE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Create Process	hlq.DEFINE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Create Queue“2” on page 234	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Create Storage Class	hlq.DEFINE.STGCLASS	ALTER	No check	-
Create Subscription	hlq.DEFINE.SUB	ALTER	No check	-
Create Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Delete CF Structure	hlq.DELETE.CFSTRUCT	ALTER	No check	-
Delete Channel	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Namelist	hlq.DELETE.NAMELIST	ALTER	hlq.NAMELIST.namelist	ALTER
Delete Process	hlq.DELETE.PROCESS	ALTER	hlq.PROCESS.process	ALTER
Delete Queue	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Delete Storage Class	hlq.DELETE.STGCLASS	ALTER	No check	-
Delete Subscription	hlq.DELETE.SUB	ALTER	No check	-
Delete Topic	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Inquire Archive	hlq.DISPLAY.ARCHIVE	READ	No check	-
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire Authentication Information Object Names	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire CF Structure	hlq.DISPLAY.CFSTRUCT	READ	No check	-
Inquire CF Structure Names	hlq.DISPLAY.CFSTRUCT	READ	No check	-
Inquire CF Structure Status	hlq.DISPLAY.CFSTATUS	READ	No check	-
Inquire Channel	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Authentication Records	hlq.DISPLAY.CHLAUTH	READ	No check	-
Inquire Channel Initiator	hlq.DISPLAY.CHINIT	READ	No check	-
Inquire Channel Names	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Status	hlq.DISPLAY.CHSTATUS	READ	No check	-
Inquire Cluster Queue Manager	hlq.DISPLAY.CLUSQMGR	READ	No check	-

Table 50. PCF commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Inquire Connection	hlq.DISPLAY.CONNPCF	READ	No check	-
Inquire Group	hlq.DISPLAY.GROUP	READ	No check	-
Inquire Log	hlq.DISPLAY.LOG	READ	No check	-
Inquire Namelist	hlq.DISPLAY.NAMELIST	READ	No check	-
Inquire Namelist Names	hlq.DISPLAY.NAMELIST	READ	No check	-
Inquire Process	hlq.DISPLAY.PROCESS	READ	No check	-
Inquire Process Names	hlq.DISPLAY.PROCESS	READ	No check	-
Inquire Pub/Sub Status	hlq.DISPLAY.PUBSUB	READ	No check	-
Inquire Queue	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Manager	hlq.DISPLAY.QMGR	READ	No check	-
Inquire Queue Names	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Status	hlq.DISPLAY.QSTATUS	READ	No check	-
Inquire Security	hlq.DISPLAY.SECURITY	READ	No check	-
Inquire SMDS	hlq.DISPLAY.SMDS	READ	No check	-
Inquire SMDSCONN	hlq.DISPLAY.SMDSCONN	READ	No check	-
Inquire Storage Class	hlq.DISPLAY.STGCLASS	READ	No check	-
Inquire Storage Class Names	hlq.DISPLAY.STGCLASS	READ	No check	-
Inquire Subscription	hlq.INQUIRE.SUB	READ	No check	-
Inquire Subscription Status	hlq.INQUIRE.SBSTATUS	READ	No check	-
Inquire System	hlq.DISPLAY.SYSTEM	READ	No check	-
Inquire Topic	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Names	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Status	hlq.DISPLAY.TPSTATUS	READ	No check	-
Inquire Usage	hlq.DISPLAY.USAGE	READ	No check	-
Move Queue	hlq.MOVE.QLOCAL	ALTER	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	ALTER
Ping Channel	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Recover CF Structure	hlq.RECOVER.CFSTRUCT	CONTROL	No check	-
Refresh Cluster	hlq.REFRESH.CLUSTER	ALTER	No check	-
Refresh Queue Manager	hlq.REFRESH.QMGR	ALTER	No check	-
Refresh Security	hlq.REFRESH.SECURITY	ALTER	No check	-
Reset CF Structure	hlq.RESET.CFSTRUCT	CONTROL	No check	-
Reset Channel	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL

Table 50. PCF commands, profiles, and their access levels (continued)

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Reset Cluster	hlq.RESET.CLUSTER	CONTROL	No check	-
Reset Queue Manager	hlq.RESET.QMGR	CONTROL	No check	-
Reset Queue Statistics	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
Reset SMDS	hlq.RESET.SMDS	CONTROL	No check	-
Resolve Channel	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Resume Queue Manager	hlq.RESUME.QMGR	CONTROL	No check	-
Resume Queue Manager Cluster	hlq.RESUME.QMGR	CONTROL	No check	-
Reverify Security	hlq.RVERIFY.SECURITY	ALTER	No check	-
Set Archive	hlq.SET.ARCHIVE	CONTROL	No check	-
Set Channel Authentication Record	hlq.SET.CHLAUTH	CONTROL	No check	-
Set Log	hlq.SET.LOG	CONTROL	No check	-
Set System	hlq.SET.SYSTEM	CONTROL	No check	-
Start Channel	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Start Channel Initiator	hlq.START.CHINIT	CONTROL	No check	-
Start Channel Listener	hlq.START.LISTENER	CONTROL	No check	-
Start SMDS Connection	hlq.START.SMDSCONN	CONTROL	No check	-
Stop Channel	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Stop Channel Initiator	hlq.STOP.CHINIT	CONTROL	No check	-
Stop Channel Listener	hlq.STOP.LISTENER	CONTROL	No check	-
Stop SMDS Connection	hlq.STOP.SMDSCONN	CONTROL	No check	-
Suspend Queue Manager	hlq.SUSPEND.QMGR	CONTROL	No check	-
Suspend Queue Manager Cluster	hlq.SUSPEND.QMGR	CONTROL	No check	-

Notes:

1. The **hlq.TOPIC.topic** resource refers to the Topic object derived from the TOPICSTR. For more details, see "Segurança de Publicação/Assinatura" on page 492
2. Setting the queue attribute STREAMQ to a non blank value also requires ALTER access level to MQADMIN or MXADMIN for `hlq.ALTER.streamQ`.

See "IBM MQ Console - required command security profiles" on page 234 for details of the IBM MQ PCF profiles required, when using the IBM MQ Console.

z/OS *IBM MQ Console - required command security profiles*

Operations performed in the IBM MQ Console by a user in the `MQWebAdmin`, or `MQWebAdminRO`, role take place under the security context of the mqweb server started task user ID. If you want to use the IBM MQ Console, the mqweb server started task user ID needs authorization to issue certain PCF commands.

Table 51 on page 235 shows, for each IBM MQ PCF command, the command security profiles required, and the corresponding access level for each profile in the MQCMDS class needed by the IBM MQ Console.

Table 51. IBM MQ Console PCF commands, profiles, and their access levels

Command	Command profile for MQCMDS	Access level for MQCMDS	Command resource profile for MQADMIN or MXADMIN	Access level for MQADMIN or MXADMIN
Change Authentication Information Object	hlq.ALTER.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Change Channel	hlq.ALTER.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Change Queue	hlq.ALTER.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Change Queue Manager	hlq.ALTER.QMGR	ALTER	No check	-
Change Topic	hlq.ALTER.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Clear Queue	hlq.CLEAR.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
Create Authentication Information Object	hlq.DEFINE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Create Channel	hlq.DEFINE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Create Queue	hlq.DEFINE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Create Subscription	hlq.DEFINE.SUB	ALTER	No check	-
Create Topic	hlq.DEFINE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Delete Authentication Information Object	hlq.DELETE.AUTHINFO	ALTER	hlq.AUTHINFO.resourcename	ALTER
Delete Channel	hlq.DELETE.CHANNEL	ALTER	hlq.CHANNEL.channel	ALTER
Delete Queue	hlq.DELETE.QUEUE	ALTER	hlq.QUEUE.queue	ALTER
Delete Subscription	hlq.DELETE.SUB	ALTER	No check	-
Delete Topic	hlq.DELETE.TOPIC	ALTER	hlq.TOPIC.topic	ALTER
Inquire Authentication Information Object	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire Authentication Information Object Names	hlq.DISPLAY.AUTHINFO	READ	No check	-
Inquire Channel	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Authentication Records	hlq.DISPLAY.CHLAUTH	READ	No check	-
Inquire Channel Initiator	hlq.DISPLAY.CHINIT	READ	No check	-
Inquire Channel Names	hlq.DISPLAY.CHANNEL	READ	No check	-
Inquire Channel Status	hlq.DISPLAY.CHSTATUS	READ	No check	-
Inquire Queue	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Manager	hlq.DISPLAY.QMGR	READ	No check	-
Inquire Queue Names	hlq.DISPLAY.QUEUE	READ	No check	-
Inquire Queue Status	hlq.DISPLAY.QSTATUS	READ	No check	-
Inquire Subscription	hlq.INQUIRE.SUB	READ	No check	-

Table 51. IBM MQ Console PCF commands, profiles, and their access levels (continued)				
Command	**Command profile for MQCMDS**	**Access level for MQCMDS**	**Command resource profile for MQADMIN or MXADMIN**	**Access level for MQADMIN or MXADMIN**
Inquire Subscription Status	hlq.INQUIRE.SBSTATUS	READ	No check	-
Inquire Topic	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Names	hlq.DISPLAY.TOPIC	READ	No check	-
Inquire Topic Status	hlq.DISPLAY.TPSTATUS	READ	No check	-
Ping Channel	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Refresh Cluster	hlq.REFRESH.CLUSTER	ALTER	No check	-
Refresh Security	hlq.REFRESH.SECURITY	ALTER	No check	-
Reset Channel	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Resolve Channel	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Set Channel Authentication Record	hlq.SET.CHLAUTH	CONTROL	No check	-
Start Channel	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Stop Channel	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL

z/OS *Profiles for command resource security*

If you have not defined the command resource security switch profile, because you want security checking for resources associated with commands, you must add resource profiles for each resource to the appropriate class. The same security profiles control both MQSC and PCF commands.

If you have not defined the command resource security switch profile, `hlq.NO.CMD.RESC.CHECKS`, because you want security checking for resources associated with commands, you must:

- Add a resource profile in the **MQADMIN** class, if using uppercase profiles, for each resource.
- Add a resource profile in the **MXADMIN** class, if using mixed case profiles, for each resource.

The same security profiles control both MQSC and PCF commands.

Profiles for command resource security checking take the form:

```
hlq.type.resourcename
```

where `hlq` can be either `qmgr-name` (queue manager name) or `qsg-name` (queue sharing group name).

A profile prefixed by the queue manager name controls access to the resources associated with commands on that queue manager. A profile prefixed by the queue sharing group name controls access to the resources associated with commands on all queue managers within the queue sharing group. This access can be overridden on an individual queue manager by defining a queue manager level profile for that command resource on that queue manager.

If your queue manager is a member of a queue sharing group and you are using both queue manager and queue sharing group level security, IBM MQ checks for a profile prefixed by the queue manager name first. If it does not find one, it looks for a profile prefixed by the queue sharing group name.

For example, the RACF profile name for command resource security checking against the model queue CREDIT.WORTHY in subsystem CSQ1 is:

```
CSQ1.QUEUE.CREDIT.WORTHY
```

Because the profiles for all types of command resource are held in the MQADMIN class, the "type" part of the profile name is needed in the profile to distinguish between resources of different types that have the same name. The "type" part of the profile name can be CHANNEL, QUEUE, TOPIC, PROCESS, or NAMELIST. For example, a user might be authorized to define hlq.QUEUE.PAYROLL.ONE, but not authorized to define hlq.PROCESS.PAYROLL.ONE

If the resource type is a queue, and the profile is a queue sharing group level profile, it controls access to one or more local queues within the queue sharing group, or access to a single shared queue from any queue manager in the queue sharing group.

MQSC commands, profiles, and their access levels shows, for each IBM MQ MQSC command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

PCF commands, profiles, and their access levels shows, for each IBM MQ PCF command, the profiles required for command security checking to be carried out, and the corresponding access level for each profile in the MQCMDS class.

z/OS *Command resource security checking for alias queues and remote queues*

Alias queue and remote queues both provide indirection to another queue. Additional points apply when you consider security checking for these queues.

Alias queues

When you define an alias queue, command resource security checks are only performed against the name of the alias queue, not against the name of the target queue to which the alias resolves.

Alias queues can resolve to both local and remote queues. If you do not want to permit users access to certain local or remote queues, you must do both of the following:

1. Do not allow the users access to these local and remote queues.
2. Restrict the users from being able to define aliases for these queues. That is, prevent them from being able to issue DEFINE QALIAS and ALTER QALIAS commands.

Remote queues

When you define a remote queue, command resource security checks are performed only against the name of the remote queue. No checks are performed against the names of the queues specified in the RNAME or XMITQ attributes in the remote queue object definition.

z/OS O perfil de segurança RESLEVEL

É possível definir um perfil especial na classe MQADMIN ou MXADMIN para controlar o número de IDs de usuário verificados para a segurança do recurso da API. Esse perfil é chamado de perfil RESLEVEL. Como este perfil afeta a segurança do recurso da API depende de como você acessa o IBM MQ.

Quando um aplicativo tenta se conectar ao IBM MQ, IBM MQ verifica o acesso que o ID do usuário associado à conexão tem para um perfil na classe MQADMIN ou MXADMIN, chamado:

```
hlq.RESLEVEL
```

Em que `hlq` pode ser `ssid` (ID do subsistema) ou `qsg` (ID do grupo de filas compartilhadas).

Os IDs de usuário associados a cada tipo de conexão são:

- O ID do usuário da tarefa conectada para conexões em batch;
- O ID do usuário do espaço de endereço do CICS para conexões do CICS
- O ID do usuário do espaço de endereço da região do IMS para conexões do IMS
- O ID do usuário do espaço de endereçamento do iniciador do canal das conexões do iniciador do canal.

Atenção: RESLEVEL é uma opção muito poderosa; ela pode causar o bypass de todas as verificações de segurança do recurso para uma conexão específica.

Se você não tiver um perfil RESLEVEL definido, deverá tomar cuidado para que nenhum outro perfil na classe MQADMIN corresponda a hlq.RESLEVEL. Por exemplo, se você tiver um perfil em MQADMIN chamado hlq. * * e nenhum perfil hlq.RESLEVEL , cuidado com as consequências do hlq. * * porque é usado para a verificação RESLEVEL.

Defina um perfil hlq.RESLEVEL e configure o UACC para NONE, em vez de não ter nenhum perfil RESLEVEL. Tenha o mínimo possível de usuários ou grupos na lista de acesso. Para obter detalhes sobre como auditar o acesso de RESLEVEL, consulte “Auditing considerations on z/OS” na página 262.

Se estiver usando somente a segurança no nível do gerenciador de filas, o IBM MQ executa verificações RESLEVEL com relação ao perfil `qmgr-name.RESLEVEL`. Se você estiver usando a segurança no nível do grupo de compartilhamento de filas apenas, o IBM MQ executa verificações RESLEVEL em relação ao Perfil `doqsg-name.RESLEVEL` . Se você estiver usando uma combinação de segurança no nível do grupo de filas compartilhadas e do gerenciador de filas, o IBM MQ primeiro verificará a existência de um perfil RESLEVEL no nível do gerenciador de filas. Se não localizar um, ele verificará se há um perfil RESLEVEL no nível do grupo de filas compartilhadas.

Se ele não puder localizar um perfil RESLEVEL, o IBM MQ permite a verificação do ID do job e da tarefa (ou usuário alternativo) para um CICS ou uma conexão do IMS. Para uma conexão em lote, o IBM MQ permite a verificação do ID de usuário da tarefa (ou alternativo). Para o inicializador de canais, o IBM MQ permite a verificação do ID do usuário do canal e o ID do usuário do MCA (ou alternativo).

Se houver um perfil RESLEVEL, o nível de verificação dependerá do ambiente e do nível de acesso para o perfil.

Lembre-se de que se o gerenciador de filas for um membro de um grupo de filas compartilhadas e você não definir esse perfil no nível do gerenciador de filas, poderá haver um definido no nível do grupo de filas compartilhadas que afetará o nível de verificação Para ativar a verificação de dois IDs de usuário, você define um perfil RESLEVEL (prefixado com o nome do gerenciador de filas do nome do grupo de filas compartilhadas) com um UACC (NONE) e assegura que os usuários relevantes não tenham acesso concedido a esse perfil.

Ao considerar o acesso que o ID do usuário do inicializador de canais tem para o RESLEVEL, lembre-se de que a conexão estabelecida pelo inicializador de canais também é a conexão usada pelos canais. Uma configuração que causa bypass de todas as verificações de segurança do recurso para o ID do usuário do inicializador de canais efetua bypass efetivamente das verificações de segurança para todos os canais. Se o acesso do ID do usuário do inicializador de canais para o RESLEVEL for diferente de NONE, apenas um ID do usuário (para um nível de acesso READ ou UPDATE) ou nenhum ID do usuário (para um nível de acesso CONTROL ou ALTER) será verificado quanto ao acesso. Se você conceder ao ID do usuário do inicializador de canais um nível de acesso diferente de NONE para o RESLEVEL, certifique-se de que entender o efeito dessa definição sobre as verificações de segurança realizadas para canais.

O uso do perfil RESLEVEL significa que registros de auditoria de segurança normais não são tomados. Por exemplo, se você colocar UAUDIT em um usuário, o acesso ao perfil hlq.RESLEVEL em MQADMIN não será auditorado.

Se você usar a opção RACF WARNING no perfil hlq.RESLEVEL, nenhuma mensagem de aviso do RACF será produzida para perfis na classe RESLEVEL.

A verificação de segurança para mensagens de relatório como CODs é controlada pelo perfil RESLEVEL associado ao aplicativo de origem. Por exemplo, se o ID do usuário de uma tarefa em lote tiver autoridade CONTROL ou ALTER para um perfil RESLEVEL, então será efetuado bypass de toda a verificação de recursos executada pela tarefa em lote, incluindo a verificação de segurança de mensagens de relatório.

Se você alterar o perfil RESLEVEL, os usuários deverão desconectar e conectar novamente antes que a mudança ocorra. (Isso inclui parar e reiniciar o inicializador de canais se o acesso que o ID do usuário do espaço de endereço de enfileiramento distribuído tiver para o perfil RESLEVEL for alterado.)

Para desativar a auditoria de RESLEVEL use o parâmetro do sistema RESAUDIT.

z/OS *RESLEVEL and batch connections*

By default, when an IBM MQ resource is being accessed through batch and batch-type connections, the user must be authorized to access that resource for the particular operation. You can bypass the security check by setting up an appropriate RESLEVEL definition.

Whether the user is checked or not is based on the user ID used at connect time, the same user ID used for the connection check.

For example, you can set up RESLEVEL so that when a user you trust accesses certain resources through a batch connection, no API-resource security checks are done; but when a user you do not trust tries to access the same resources, security checks are carried out as normal. You should set up RESLEVEL checking to bypass API-resource security checks only when you sufficiently trust the user and the programs run by that user.

The following table shows the checks made for batch connections.

Table 52. Checks made at different RACF access levels for batch connections

RACF access level	**Level of checking**
NONE	Resource checks performed
READ	Resource checks performed
UPDATE	Resource checks performed
CONTROL	No check.
ALTER	No check.

z/OS *RESLEVEL and system functions*

The application of RESLEVEL to the operation and control panels, and to CSQUTIL.

The operation and control panels and the CSQUTIL utility are batch-type applications that make requests to the queue manager's command server, and so they are subject to the considerations described in “RESLEVEL and batch connections” on page 239. You can use RESLEVEL to bypass security checking for the SYSTEM.COMMAND.INPUT and SYSTEM.COMMAND.REPLY.MODEL queues that they use, but not for the dynamic queues SYSTEM.CSQXCMD.*, SYSTEM.CSQOREXX.*, and SYSTEM.CSQUTIL.*.

The command server is an integral part of the queue manager and so does not have connection or RESLEVEL checking associated with it. To maintain security, therefore, the command server must confirm that the user ID of the requesting application has authority to open the queue being used for replies. For the operations and control panels, this is SYSTEM.CSQOREXX.*. For CSQUTIL, it is SYSTEM.CSQUTIL.*. Users must be authorized to use these queues, as described in “System queue security” on page 213, in addition to any RESLEVEL authorization they are given.

For other applications using the command server, it is the queue they name as their reply-to queue. Such other applications might deceive the command server into placing messages on unauthorized queues by passing (in the message context) a more trusted user ID than its own to the command server. To prevent this, use a CONTEXT profile to protect the identity context of messages placed on SYSTEM.COMMAND.INPUT.

z/OS *RESLEVEL and CICS connections*

By default, when an API-resource security check is made on a CICS connection, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

The first user ID checked is that of the CICS address space. This is the user ID on the job card of the CICS job, or the user ID assigned to the CICS started task by the z/OS STARTED class or the started procedures table. (It is not the CICS DFLTUSER.)

The second user ID checked is the user ID associated with the CICS transaction.

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED. Both the CICS address space user ID and the user ID of the person running the CICS transaction must have access to the resource at the correct level.

How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested. See Table 53 on page 240 for more information.

The user IDs checked depend on the user ID used at connection time, that is, the CICS address space user ID. This control enables you to bypass API-resource security checking for IBM MQ requests coming from one system (for example, a test system, TESTCICS,) but to implement them for another (for example, a production system, PRODCICS).

Note: If you set up your CICS address space user ID with the "trusted" attribute in the STARTED class or the RACF started procedures table ICHRIN03, this overrides any user ID checks for the CICS address space established by the RESLEVEL profile for your queue manager (that is, the queue manager does not perform the security checks for the CICS address space). For more information, see Securing CICS.

The following table shows the checks made for CICS connections.

Table 53. Checks made at different RACF access levels for CICS connections

RACF access level	Level of checking
NONE	IBM MQ checks the CICS address space user ID and the transaction user ID.
READ	IBM MQ checks the CICS address space user ID only.
UPDATE	If the transaction is defined to CICS with RESSEC(YES), IBM MQ checks the CICS address space user ID and the transaction user ID.
UPDATE	If the transaction is defined to CICS with RESSEC(NO), IBM MQ checks the CICS address space user ID only.
CONTROL or ALTER	IBM MQ does not check any user IDs.

z/OS *RESLEVEL and IMS connections*

By default, when an API-resource security check is made for an IMS connection, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

By default, when an API-resource security check is made for an IMS connection, two user IDs are checked to see if access is allowed to the resource.

The first user ID checked is that of the address space of the IMS region. This is taken from either the USER field from the job card or the user ID assigned to the region from the z/OS STARTED class or the started procedures table (SPT).

The second user ID checked is associated with the work being done in the dependent region. It is determined according to the type of the dependent region as shown in How the second user ID is determined for the IMS(tm) connection.

If either the first or second IMS user ID does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED.

The setting of IBM MQ RESLEVEL profiles cannot alter the user ID under which IMS transactions are scheduled from the IBM-supplied MQ-IMS trigger monitor program CSQQTRMN. This user ID is the PSBNAME of that trigger monitor, which by default is CSQQTRMN.

How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested. The possible checks are:

- Check the IMS region address space user ID and the second user ID or alternate user ID.
- Check IMS region address space user ID only.
- Do not check any user IDs.

The following table shows the checks made for IMS connections.

Table 54. Checks made at different RACF access levels for IMS connections

RACF access level	Level of checking
NONE	Check the IMS address space user ID and the IMS second user ID or alternate user ID.
READ	Check the IMS address space user ID.
UPDATE	Check the IMS address space user ID.
CONTROL	No check.
ALTER	No check.

z/OS *RESLEVEL and the channel initiator connection*

By default, when an API-resource security check is made by the channel initiator, two user IDs are checked. You can change which user IDs are checked by setting up a RESLEVEL profile.

By default, when an API-resource security check is made by the channel initiator, two user IDs are checked to see if access is allowed to the resource.

The user IDs checked can be that specified by the MCAUSER channel attribute, that received from the network, that of the channel initiator address space, or the alternate user ID for the message descriptor. Which user IDs are checked depends on the communication protocol you are using and the setting of the PUTAUT channel attribute. See “User IDs used by the channel initiator” on page 246 for more information.

If one of these user IDs does not have access to the resource, the request fails with a completion code of MQRC_NOT_AUTHORIZED.

How RESLEVEL can affect the checks made

Depending on how you set up your RESLEVEL profile, you can change which user IDs are checked when access to a resource is requested, and how many are checked.

The following table shows the checks made for the channel initiator's connection, and for all channels since they use this connection.

Table 55. Checks made at different RACF access levels for channel initiator connections

RACF access level	Level of checking
NONE	Check two user IDs.
READ	Check one user ID.
UPDATE	Check one user ID.
CONTROL	No check.
ALTER	No check.
Note: See “User IDs used by the channel initiator” on page 246 for a definition of the user IDs checked	

z/OS RESLEVEL and intra-group queuing

By default, when an API-resource security check is made by the intra-group queuing agent, two user IDs are checked to see if access is allowed to the resource. You can change which user IDs are checked by setting up an RESLEVEL profile.

The user IDs checked can be the user ID determined by the IGQUSER attribute of the receiving queue manager, the user ID of the queue manager within the queue sharing group that put the message on to the SYSTEM.QSG.TRANSMIT.QUEUE, or the alternate user ID specified in the *UserIdentifier* field of the message descriptor of the message. See "User IDs used by the intra-group queuing agent" on page 250 for more information.

Because the intra-group queuing agent is an internal queue manager task, it does not issue an explicit connect request and runs under the user ID of the queue manager. The intra-group queuing agent starts at queue manager initialization. During the initialization of the intra-group queuing agent, IBM MQ checks the access that the user ID associated with the queue manager has to a profile in the MQADMIN class called:

```
hlq.RESLEVEL
```

This check is always performed unless the hlq.NO.SUBSYS.SECURITY switch has been set.

If there is no RESLEVEL profile, IBM MQ enables checking for two user IDs. If there is a RESLEVEL profile, the level of checking depends on the access level granted to the user ID of the queue manager for the profile. Checks made at different RACF(r) access levels for the intra-group queuing agent shows the checks made for the intra-group queuing agent.

Table 56. Checks made at different RACF access levels for the intra-group queuing agent

RACF access level	**Level of checking**
NONE	Check two user IDs.
READ	Check one user ID.
UPDATE	Check one user ID.
CONTROL	No check.
ALTER	No check.
Note: See "User IDs used by the intra-group queuing agent" on page 250 for a definition of the user IDs checked	

If the permissions granted to the RESLEVEL profile for the queue manager's user ID are changed, the intra-group queuing agent must be stopped and restarted to pick up the new permissions. Because there is no way to independently stop and restart the intra-group queuing agent, the queue manager must be stopped and restarted to achieve this.

z/OS RESLEVEL and the user IDs checked

Example of setting a RESLEVEL profile and granting access to it.

User ID checking against profile name for batch connections through User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels show how RESLEVEL affects which user IDs are checked for different MQI requests.

For example, you have a queue manager called QM66 with the following requirements:

- User WS21B is to be exempt from resource security.
- CICS started task WXNCICS running under address space user ID CICSWXN is to perform full resource checking only for transactions defined with RESSEC(YES).

To define the appropriate RESLEVEL profile, issue the following RACF command:

```
RDEFINE MQADMIN QM66.RESLEVEL UACC(NONE)
```

Then give the users access to this profile, using the following commands:

```
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(WS21B) ACCESS(CONTROL)
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(CICSWXN) ACCESS(UPDATE)
```

If you make these changes while the user IDs are connected to queue manager QM66, the users must disconnect and connect again before the change takes place.

If subsystem security is not active when a user connects but, while this user is still connected, subsystem security becomes active, full resource security checking is applied to the user. The user must reconnect to get the correct RESLEVEL processing.

z/OS User IDs for security checking on z/OS

IBM MQ initiates security checks based on user IDs associated with users, terminals, applications, and other resources. This collection of topics lists which user IDs are used for each type of security check.

z/OS *User IDs for connection security*

The user ID used for connection security depends on the type of connection.

Connection type	User ID contents
Batch connection	The user ID of the connecting task. For example: • The TSO user ID • The user ID assigned to a batch job by the USER JCL parameter • The user ID assigned to a started task by the STARTED class or the started procedures table
CICS connection	The CICS address space user ID.
IMS connection	The IMS region address space user ID.
Channel initiator connection	The channel initiator address space user ID.

z/OS *User IDs for command and command resource security*

The user ID used for command security or command resource security depends on where the command is issued from.

Issued from...	User ID contents
CSQINP1, CSQINP2, or CSQINPT	No check is made.
System command input queue	The user ID found in the *UserIdentifier* of the message descriptor of the message that contains the command. If the message does not contain a *UserIdentifier*, a user ID of blanks is passed to the security manager.
Console	The user ID signed onto the console. If the console is not signed on, the default user ID set by the CMDUSER system parameter in CSQ6SYSP. To issue commands from a console, the console must have the z/OS SYS AUTHORITY attribute.
SDSF/TSO console	TSO or job user ID.

Issued from...	User ID contents
Operations and control panels	TSO user ID. If you are going to use the operations and control panels, you must have the appropriate authority to issue the commands corresponding to the actions that you choose. In addition, you must have READ access to all the hlq.DISPLAY. *object* profiles in the MQCMDS class because the panels use the various DISPLAY commands to gather the information that they present.
MGCRE	If MGCRE is used with UTOKEN, the user ID in the UTOKEN. If MGCRE is issued without the UTOKEN, the TSO or job user ID is used.
CSQ0UTIL	Job user ID.
CSQUTIL	Job user ID.
CSQINPX	User ID of the channel initiator address space.

z/OS *User IDs for resource security (MQOPEN, MQSUB, and MQPUT1)*

This information shows the contents of the user IDs for normal and alternate user IDs for each type of connection. The number of checks is defined by the RESLEVEL profile. The user ID checked is that used for **MQOPEN**, **MQSUB**, or **MQPUT1** calls.

Note: All user ID fields are checked exactly as they are received. No conversions take place, and, for example, three user ID fields containing "Bob", "BOB", and "bob" are not equivalent.

z/OS *User IDs checked for batch connections*

The user ID checked for a batch connection depends on how the task is run and whether an alternate user ID has been specified.

Table 57. User ID checking against profile name for batch connections

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
No	-	JOB	JOB
Yes	JOB	JOB	ALT

Key:

ALT
Alternate user ID.

JOB

- The user ID of a TSO or z/OS UNIX System Services sign-on.
- The user ID assigned to a batch job.
- The user ID assigned to a started task by the STARTED class or the started procedures table.
- The user ID associated with the executing Db2 stored procedure

A Batch job is performing an MQPUT1 to a queue called Q1 with RESLEVEL set to READ and alternate user ID checking turned off.

Checks made at different RACF(r) access levels for batch connections and User ID checking against profile name for batch connections show that the job user ID is checked against profile hlq.Q1.

z/OS *User IDs checked for CICS connections*

The user IDs checked for CICS connections depend on whether one or two checks are to be carried out, and whether an alternate user ID is specified.

Table 58. User ID checking against profile name for CICS-type user IDs

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
No, 1 check	-	ADS	ADS
No, 2 checks	-	ADS+TXN	ADS+TXN
Yes, 1 check	ADS	ADS	ADS
Yes, 2 checks	ADS+TXN	ADS+TXN	ADS+ALT

Key:

ALT
Alternate user ID

ADS
The user ID associated with the CICS batch job or, if CICS is running as a started task, through the STARTED class or the started procedures table.

TXN
The user ID associated with the CICS transaction. This is normally the user ID of the terminal user who started the transaction. It can be the CICS DFLTUSER, a PRESET security terminal, or a manually signed-on user.

Determine the user IDs checked for the following conditions:

- The RACF access level to the RESLEVEL profile, for a CICS address space user ID, is set to NONE.
- An MQOPEN call is made against a queue with MQOO_OUTPUT and MQOO_PASS_IDENTITY_CONTEXT.

First, see how many CICS user IDs are checked based on the CICS address space user ID access to the RESLEVEL profile. From Table 53 on page 240 in topic “RESLEVEL and CICS connections” on page 239, two user IDs are checked if the RESLEVEL profile is set to NONE. Then, from Table 58 on page 245 on, these checks are carried out:

- The hlq.ALTERNATE.USER.userid profile is not checked.
- The hlq.CONTEXT.queuename profile is checked with both the CICS address space user ID and the CICS transaction user ID.
- The hlq.resourcename profile is checked with both the CICS address space user ID and the CICS transaction user ID.

This means that four security checks are made for this MQOPEN call.

z/OS *User IDs checked for IMS connections*

The user IDs checked for IMS connections depend on whether one or two checks are to be performed, and whether an alternate user ID is specified. If a second user ID is checked, it depends on the type of dependent region and on which user IDs are available.

Table 59. User ID checking against profile name for IMS-type user IDs

Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuenam e profile	hlq.resourcename profile
No, 1 check	-	REG	REG
No, 2 checks	-	REG+SEC	REG+SEC
Yes, 1 check	REG	REG	REG

Table 59. User ID checking against profile name for IMS-type user IDs (continued)			
Alternate user ID specified on open?	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
Yes, 2 checks	REG+SEC	REG+SEC	REG+ALT

Key:

ALT
Alternate user ID.

REG
The user ID is normally set through the STARTED class or the started procedures table or, if IMS is running, from a submitted job, by the USER JCL parameter.

SEC
The second user ID is associated with the work being done in a dependent region. It is determined according to Table 60 on page 246.

Table 60. How the second user ID is determined for the IMS connection	
Types of dependent region	**Hierarchy for determining the second user ID**
• BMP message driven and successful GET UNIQUE issued. • IFP and GET UNIQUE issued. • MPP.	User ID associated with the IMS transaction if the user is signed on. LTERM name if available. PSBNAME.
• BMP message driven and successful GET UNIQUE not issued. • BMP not message driven. • IFP and GET UNIQUE not issued.	User ID associated with the IMS dependent region address space if this is not all blanks or all zeros. PSBNAME.

z/OS *User IDs used by the channel initiator*

This collection of topics describes the user IDs used and checked for receiving channels and for client MQI requests issued over server-connection channels. Information is provided for TCP/IP and for LU6.2

You can use the PUTAUT parameter of the receiving channel definition to determine the type of security checking used. To get consistent security checking throughout your IBM MQ network, you can use the ONLYMCA and ALTMCA options.

You can use the DISPLAY CHSTATUS command to determine the user identifier used by the MCA.

z/OS *Receiving channels using TCP/IP*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

Table 61. User IDs checked against profile name for TCP/IP channels			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
DEF, 1 check	-	CHL	CHL
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL

Table 61. User IDs checked against profile name for TCP/IP channels (continued)			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	-	MCA	MCA
ONLYMCA, 2 checks	-	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

CHL (Channel user ID)
On TCP/IP, security is not supported by the communication system for the channel. If Transport Layer Security (TLS) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by using RACF Certificate Name Filtering (CNF), is used. If no associated user ID is found, or if TLS is not being used, the user ID of the channel initiator address space of the receiver or requester end is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

Note: The use of RACF Certificate Name Filtering (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote user across the world, and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the receiver or requester is used. This also applies to TCP/IP channels using TLS.

ALT (Alternate user ID)
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an **MQOPEN** or **MQPUT1** call is issued for the target destination queue.

z/OS *Receiving channels using LU 6.2*

The user IDs checked depend on the PUTAUT option of the channel and on whether one or two checks are to be performed.

Table 62. User IDs checked against profile name for LU 6.2 channels			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
DEF, 1 check	-	CHL	CHL
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
CTX, 2 checks	CHL + MCA	CHL + MCA	CHL + ALT

Table 62. User IDs checked against profile name for LU 6.2 channels (continued)			
PUTAUT option specified on receiver or requester channel	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
ONLYMCA, 1 check	-	MCA	MCA
ONLYMCA, 2 checks	-	MCA	MCA
ALTMCA, 1 check	MCA	MCA	MCA
ALTMCA, 2 checks	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the receiver; if blank, the channel initiator address space user ID of the receiver or requester side is used.

CHL (Channel user ID)

Requester-server channels
If the channel is started from the requester, there is no opportunity to receive a network user ID (the channel user ID).

If the PUTAUT parameter is set to DEF or CTX on the requester channel, the channel user ID is that of the channel initiator address space of the requester because no user ID is received from the network.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA, the channel user ID is ignored and the MCA user ID of the requester is used.

Other channel types
If the PUTAUT parameter is set to DEF or CTX on the receiver or requester channel, the channel user ID is the user ID received from the communications system when the channel is initiated.

- If the sending channel is on z/OS, the channel user ID received is the channel initiator address space user ID of the sender.
- If the sending channel is on a different platform (for example, AIX), the channel user ID received is typically provided by the USERID parameter of the channel definition.

If the user ID received is blank, or no user ID is received, a channel user ID of blanks is used.

ALT (Alternate user ID)
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object descriptor before an MQOPEN or MQPUT1 call is issued for the target destination queue.

z/OS *Client MQI requests*

Various user IDs can be used, depending on which user IDs and environment variables have been set. These user IDs are checked against various profiles, depending on the PUTAUT option used and whether an alternate user ID is specified.

This section describes the user IDs checked for client MQI requests issued over server-connection channels for TCP/IP and LU 6.2. The MCA user ID and channel user ID are as for the TCP/IP and LU 6.2 channels described in the previous sections.

For server-connection channels, the user ID received from the client is used if the MCAUSER attribute is blank.

See “Controle de acesso para clientes” on page 107 for more information.

For client **MQOPEN**, **MQSUB**, and **MQPUT1** requests, use the following rules to determine the profile that is checked:

- If the request specifies alternate-user authority, a check is made against the *hlq*.ALTERNATE.USER. *userid* profile.
- If the request specifies context authority, a check is made against the *hlq*.CONTEXT. *queuename* profile.
- For all **MQOPEN**, **MQSUB**, and **MQPUT1** requests, a check is made against the *hlq.resourcename* profile.

When you have determined which profiles are checked, use the following table to determine which user IDs are checked against these profiles.

Table 63. User IDs checked against profile name for LU 6.2 and TCP/IP server-connection channels

PUTAUT option specified on server-connection channel	Alternate user ID specified on open?	hlq.ALTERNATE.USER.userid profile	hlq.CONTEXT.queuename profile	hlq.resourcename profile
DEF, 1 check	No	-	CHL	CHL
DEF, 1 check	Yes	CHL	CHL	CHL
DEF, 2 checks	No	-	CHL + MCA	CHL + MCA
DEF, 2 checks	Yes	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 check	No	-	MCA	MCA
ONLYMCA, 1 check	Yes	MCA	MCA	MCA
ONLYMCA, 2 checks	No	-	MCA	MCA
ONLYMCA, 2 checks	Yes	MCA	MCA	MCA + ALT

Key:

MCA (MCA user ID)
The user ID specified for the MCAUSER channel attribute at the server-connection; if blank, the channel initiator address space user ID is used.

CHL (Channel user ID)
On TCP/IP, security is not supported by the communication system for the channel. If Transport Layer Security (TLS) is being used and a digital certificate has been flowed from the partner, the user ID associated with this certificate (if installed), or the user ID associated with a matching filter found by using RACF Certificate Name Filtering (CNF), is used. If no associated user ID is found, or if TLS is not being used, the user ID of the channel initiator address space is used as the channel user ID on channels defined with the PUTAUT parameter set to DEF or CTX.

Note: The use of RACF Certificate Name Filtering (CNF) allows you to assign the same RACF user ID to multiple remote users, for example all the users in the same organization unit, who would naturally all have the same security authority. This means that the server does not have to have a copy of the certificate of every possible remote user across the world, and greatly simplifies certificate management and distribution.

If the PUTAUT parameter is set to ONLYMCA or ALTMCA for the channel, the channel user ID is ignored and the MCA user ID of the server-connection channel is used. This also applies to TCP/IP channels using TLS.

ALT (Alternate user ID)
The user ID from the context information (that is, the *UserIdentifier* field) within the message descriptor of the message. This user ID is moved into the *AlternateUserID* field in the object or subscription descriptor before an **MQOPEN**, **MQSUB** or **MQPUT1** call is issued on behalf of the client application.

z/OS *Channel initiator example*

An example of how user IDs are checked against RACF profiles.

A user performs an **MQPUT1** operation to a queue on queue manager QM01 that resolves to a queue called QB on queue manager QM02. The message is sent on a TCP/IP channel called QM01.TO.QM02. RESLEVEL is set to NONE, and the open is performed with alternate user ID and context checking. The receiver channel definition has PUTAUT(CTX) and the MCA user ID is set. Which user IDs are used on the receiving channel to put the message to queue QB?

Answer: Table 55 on page 241 shows that two user IDs are checked because RESLEVEL is set to NONE.

Table 61 on page 246 shows that, with PUTAUT set to CTX and 2 checks, the following user IDs are checked:

- The channel initiator user ID and the MCAUSER user ID are checked against the hlq.ALTERNATE.USER.userid profile.
- The channel initiator user ID and the MCAUSER user ID are checked against the hlq.CONTEXT.queuename profile.
- The channel initiator user ID and the alternate user ID specified in the message descriptor (MQMD) are checked against the hlq.Q2 profile.

z/OS *User IDs used by the intra-group queuing agent*

The user IDs that are checked when the intra-group queuing agent opens destination queues are determined by the values of the **IGQAUT** and **IGQUSER** queue manager attributes.

The possible user IDs are:

Intra-group queuing user ID (IGQ)
The user ID determined by the **IGQUSER** attribute of the receiving queue manager. If this is set to blanks, the user ID of the receiving queue manager is used. However, because the receiving queue manager has authority to access all queues defined to it, security checks are not performed for the receiving queue manager's user ID. In this case:

- If only one user ID is to be checked and the user ID is that of the receiving queue manager, no security checks take place. This can occur when **IGQAUT** is set to ONLYIGQ or ALTIGQ.
- If two user IDs are to be checked and one of the user IDs is that of the receiving queue manager, security checks take place for the other user ID only. This can occur when **IGQAUT** is set to DEF, CTX, or ALTIGQ.
- If two user IDs are to be checked and both user IDs are that of the receiving queue manager, no security checks take place. This can occur when **IGQAUT** is set to ONLYIGQ.

Sending queue manager user ID (SND)
The user ID of the queue manager within the queue sharing group that put the message on to the SYSTEM.QSG.TRANSMIT.QUEUE.

Alternate user ID (ALT)
The user ID specified in the *UserIdentifier* field in the message descriptor of the message.

Table 64. User IDs checked against profile name for intra-group queuing			
IGQAUT option specified on receiving queue manager	**hlq.ALTERNATE.USER.userid profile**	**hlq.CONTEXT.queuename profile**	**hlq.resourcename profile**
DEF, 1 check	-	SND	SND
DEF, 2 checks	-	SND +IGQ	SND +IGQ
CTX, 1 check	SND	SND	SND
CTX, 2 checks	SND + IGQ	SND +IGQ	SND + ALT
ONLYIGQ, 1 check	-	IGQ	IGQ
ONLYIGQ, 2 checks	-	IGQ	IGQ
ALTIGQ, 1 check	-	IGQ	IGQ
ALTIGQ, 2 checks	IGQ	IGQ	IGQ + ALT

Key:

ALT
Alternate user ID.

IGQ
IGQ user ID.

SND
Sending queue manager user ID.

z/OS *Blank user IDs and UACC levels*

If a blank user ID occurs, a RACF undefined user is signed on. Do not grant wide-ranging access to the undefined user.

Blank user IDs can exist when a user is manipulating messages using context or alternate-user security, or when IBM MQ is passed a blank user ID. For example, a blank user ID is used when a message is written to the system-command input queue without context.

Note: A user ID of " * " (that is, an asterisk character followed by seven spaces) is treated as an undefined user ID.

IBM MQ passes the blank user ID to RACF and a RACF undefined user is signed on. All security checks then use the universal access (UACC) for the relevant profile. Depending on how you have set your access levels, the UACC might give the undefined user a wide-ranging access.

For example, if you issue this RACF command from TSO:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.EVERYONE UACC(UPDATE)
```

you define a profile that enables both z/OS-defined user IDs (that have not been put in the access list) and the RACF undefined user ID to put messages on, and get messages from, that queue.

To protect against blank user IDs you must plan your access levels carefully, and limit the number of people who can use context and alternate-user security. You must prevent people using the RACF undefined user ID from getting access to resources that they must not access. However, at the same time, you must allow access to people with defined user IDs. To do this, you can specify a user ID of asterisk (*) in a RACF command PERMIT, giving access to resources for all defined user IDs. Therefore all

undefined user IDs (such as " * ") are denied access. For example, these RACF commands prevent the RACF undefined user ID from gaining access to the queue to put or get messages:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY UACC(NONE)
PERMIT Q.AVAILABLE.TO.RACF.DEFINED.USERS.ONLY CLASS(MQQUEUE) ACCESS(UPDATE) ID(*)
```

z/OS z/OS user IDs and Multi-Factor Authentication (MFA)

IBM Multi-Factor Authentication for z/OS allows z/OS security administrators to enhance SAF authentication, by requiring identified users to use multiple authentication factors (for example, both a password and a cryptographic token) to sign on to a z/OS system. IBM MFA also provides support for time-based one time password generation technologies such as RSA SecureId.

For the most part, IBM MQ is unaware of how users have "logged on" to the CICS or batch systems that are driving IBM MQ work, the signed on user ID credential is associated with the z/OS task or address space and IBM MQ uses this for checking authorization to resources. User IDs enabled for MFA can be used for authorization to IBM MQ resources and authentication through pass tickets used with the CICS and IMS bridges.

Important: Special considerations apply however, when using applications, such as the IBM MQ Explorer, which pass a user ID and password credentials on an MQCONNX API call with the *MQCSP_AUTH_USER_ID_AND_PWD* option. IBM MQ has no facility to pass an additional credential on this API request.

Limitations and potential workarounds are described in the following text.

IBM MQ Explorer

The IBM MQ Explorer cannot be used to log on to a z/OS system with a userid for which MFA is enabled because there is no facility for passing a second authentication factor from the IBM MQ Explorer to z/OS.

Additionally, there are two different mechanisms used by the IBM MQ Explorer to re-use a user ID and password credential, that need special attention when one time use passwords are in effect:

1. IBM MQ Explorer has the capability to store passwords in an obfuscated format on the local machine for login at a later time. This capability must be disabled by having explorer prompt for a password each time a connection is made to the z/OS queue manager.

   To do this, use the following procedure:

   a. Select **Queue Managers**.
   b. From the list displayed, choose the queue manager you require and right click that queue manager.
   c. Select **Connection Details** from the menu list that appears.
   d. Select **Properties** from the next menu list and choose the **Userid** tab.

      Ensure that you select the **prompt for password** radio button.
2. Various operations in the IBM MQ Explorer, such as browsing messages on queues, testing subscriptions, and so on, start a new thread which authenticates to IBM MQ using the credential first used at logon. Since the password credential cannot be re-used, you cannot use these operations.

There are two possible workarounds at the MFA configuration level for these issues:

- Use the application ID exclusion of MFA to exclude the IBM MQ tasks from MFA processing altogether.

  To do this, issue the following commands:

  1. `RDEFINE MFADEF MFABYPASS.USERID.`*`chinuser`*

     where *chinuser* is the channel initiator address space level user Id (associated with the channel initiator through the STC class)

  2. `PERMIT MFABYPASS.USERID.`*`chinuser`*` CLASS `*`MFADEF`*` ACCESS(READ) ID(`*`explorer user`*`)`

For more information on this approach, see Bypassing IBM MFA for applications.

- Use Out-of-band support on MFA, which was introduced with IBM MFA 1.2. With this approach, you pre-authenticate to the IBM MFA web server, and in addition to your user ID and password, specify additional authentication as determined through the policy. IBM MFA server generates a cache token credential that you then specify on the IBM MQ Explorer authentication dialogue. The security administrator can allow this credential to be replayed for a reasonable period of time, so enabling normal IBM MQ Explorer use.

For more information on this approach see Introduction to IBM MFA.

z/OS IBM MQ for z/OS security management

IBM MQ uses an in-storage table to hold information relating to each user and the access requests made by each user. To manage this table efficiently and to reduce the number of requests made from IBM MQ to the external security manager (ESM), a number of controls are available.

These controls are available through both the operations and control panels and IBM MQ commands.

z/OS *User ID reverification*

If the RACF definition of a user who is using IBM MQ resources has been changed, for example by connecting the user to a new group, you can tell the queue manager to sign this user on again the next time it tries to access an IBM MQ resource. You can do this by using the IBM MQ command RVERIFY SECURITY.

- User HX0804 is getting and putting messages to the PAYROLL queues on queue manager PRD1. However HX0804 now requires access to some of the PENSION queues on the same queue manager (PRD1).
- The data security administrator connects user HX0804 to the RACF group that allows access to the PENSION queues.
- So that HX0804 can access the PENSION queues immediately (that is, without shutting down queue manager PRD1 or waiting for HX0804 to time out) you must use the IBM MQ command:

```
RVERIFY SECURITY(HX0804)
```

Note: If you turn off user ID timeout for long periods of time (days or even weeks) while the queue manager is running, you must remember to run the RVERIFY SECURITY command for any users that have been revoked or deleted in that time.

z/OS *User ID timeouts*

You can make IBM MQ sign a user off a queue manager after a period of inactivity.

When a user accesses an IBM MQ resource, the queue manager tries to sign this user on to the queue manager (if subsystem security is active). This means that the user is authenticated to the ESM. This user remains signed on to IBM MQ until either the queue manager is shut down, or until the user ID is *timed out* (the authentication lapses) or reverified (reauthenticated).

When a user is timed out, the user ID is *signed off* within the queue manager and any security-related information retained for this user is discarded. The signing on and off of the user within the queue manager is not apparent to the application program or to the user.

Users are eligible for timeout when they have not used any IBM MQ resources for a predetermined amount of time. This time period is set by the MQSC ALTER SECURITY command.

Two values can be specified in the ALTER SECURITY command:

TIMEOUT
The time period in minutes that an unused user ID and its associated resources can remain within the IBM MQ queue manager.

INTERVAL
The time period in minutes between checks for user IDs and their associated resources, to determine whether the *TIMEOUT* has expired.

For example, if the *TIMEOUT* value is 30 and the *INTERVAL* value is 10, every 10 minutes IBM MQ checks user IDs and their associated resources to determine whether any have not been used for 30 minutes. If a timed-out user ID is found, that user ID is signed off within the queue manager. If any timed-out resource information associated with non-timed-out user IDs is found, that resource information is discarded. If you do not want to time out user IDs, set the *INTERVAL* value to zero. However, if the *INTERVAL* value is zero, storage occupied by user IDs and their associated resources is not freed until you issue a **REFRESH SECURITY** or **RVERIFY SECURITY** command.

Tuning this value can be important if you have many one-off users. If you set small interval and timeout values, resources that are no longer required are freed.

Note: If you use values for *INTERVAL* or *TIMEOUT* other than the defaults, you must reenter the command at every queue manager startup. You can do this automatically by putting the **ALTER SECURITY** command in the CSQINP1 data set for that queue manager.

z/OS *Refreshing queue manager security on z/OS*

IBM MQ for z/OS caches RACF data to improve performance. When you change certain security classes, you must refresh this cached information. Refresh security infrequently, for performance reasons. You can also choose to refresh only TLS security information.

When a queue is opened for the first time (or for the first time since a security refresh) IBM MQ performs a RACF check to obtain the user's access rights and places this information in the cache. The cached data includes user IDs and resources on which security checking has been performed. If the queue is opened again by the same user, the presence of the cached data means that IBM MQ does not have to issue RACF checks, which improves performance. The action of a security refresh is to discard any cached security information and so force IBM MQ to make a new check against RACF. Whenever you add, change or delete a RACF resource profile that is held in the MQADMIN, MXADMIN, MQPROC, MXPROC, MQQUEUE, MXQUEUE, MQNLIST, MXNLIST, or MXTOPIC class, you must tell the queue managers that use this class to refresh the security information that they hold. To do this, issue the following commands:

- The RACF SETROPTS RACLIST(classname) REFRESH command to refresh at the RACF level.
- The IBM MQ REFRESH SECURITY command to refresh the security information held by the queue manager. This command needs to be issued by each queue manager that accesses the profiles that have changed. If you have a queue sharing group, you can use the command scope attribute to direct the command to all the queue managers in the group.

  Note: If you have connected a new user to an existing group, you need to run the IBM MQ RVERIFY SECURITY(userid) command. The REFRESH SECURITY(*) command does not let the queue manager sign this user on again, the next time it tries to access an IBM MQ resource.

If you are using generic profiles in any of the IBM MQ classes, you must also issue normal RACF refresh commands if you change, add, or delete any generic profiles. For example, SETROPTS GENERIC(classname) REFRESH.

However, if a RACF resource profile is added, changed or deleted, and the resource to which it applies has not yet been accessed (so no information is cached), IBM MQ uses the new RACF information without a REFRESH SECURITY command being issued.

If RACF auditing is turned on, (for example, by using the RACF RALTER AUDIT(access-attempt (audit_access_level)) command), no caching takes place, and therefore IBM MQ refers directly to the RACF dataspace for every check. Changes are therefore picked up immediately and REFRESH SECURITY is not necessary to access the changes. You can confirm whether RACF auditing is on by using the RACF RLIST command. For example, you could issue the command

```
RLIST MQQUEUE (qmgr.SYSTEM.COMMAND.INPUT) GEN
```

and receive the results

```
CLASS      NAME
-----      ----
MQQUEUE    QP*.SYSTEM.COMMAND.*.** (G)
    AUDITING
    --------
    FAILURES(READ)
```

This indicates that auditing is set on. For more information, see the *z/OS Security Server RACF Auditor's Guide* and the *z/OS Security Server RACF Command Language Reference*.

Figure 17 on page 255 summarizes the situations in which security information is cached and in which cached information is used.

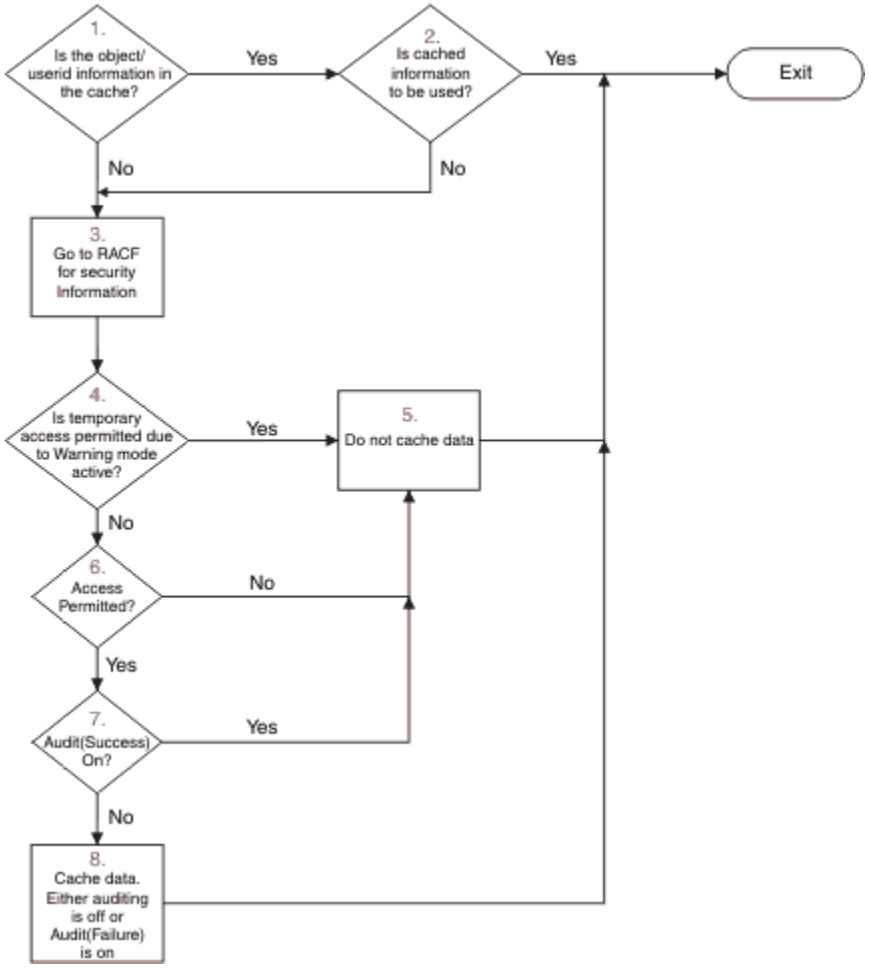


Figure 17. Logic flow for IBM MQ security caching

If you change your security settings by adding or deleting switch profiles in the MQADMIN or MXADMIN classes, use one of these commands to pick up these changes dynamically:

```
REFRESH SECURITY(*)
REFRESH SECURITY(MQADMIN)
REFRESH SECURITY(MXADMIN)
```

This means you can activate new security types, or deactivate them without having to restart the queue manager.

For performance reasons, these are the only classes affected by the REFRESH SECURITY command. You do not need to use REFRESH SECURITY if you change a profile in either the MQCONN or MQCMDS classes.

Note: A refresh of the MQADMIN or MXADMIN class is not required if you change a RESLEVEL security profile.

For performance reasons, use REFRESH SECURITY as infrequently as possible, ideally at off-peak times. You can minimize the number of security refreshes by connecting users to RACF groups that are already in the access list for IBM MQ profiles, rather than putting individual users in the access lists. In this way, you change the user rather than the resource profile. You can also RVERIFY SECURITY the appropriate user instead of refreshing security.

As an example of REFRESH SECURITY, suppose you define the new profiles to protect access to queues starting with INSURANCE.LIFE on queue manager PRMQ. You use these RACF commands:

```
RDEFINE MQQUEUE PRMQ.INSURANCE.LIFE.** UACC(NONE)
PERMIT PRMQ.INSURANCE.LIFE.** ID(LIFEGRP) ACCESS(UPDATE)
```

You must issue the following command to tell RACF to refresh the security information that it holds, for example:

```
SETROPTS RACLIST(MQQUEUE) REFRESH
```

Because these profiles are generic, you must tell RACF to refresh the generic profiles for MQQUEUE. For example:

```
SETROPTS GENERIC(MQQUEUE) REFRESH
```

Then you must use this command to tell queue manager PRMQ that the queue profiles have changed:

```
REFRESH SECURITY(MQQUEUE)
```

Refreshing SSL/TLS security

To refresh the cached view of the TLS Key Repository, issue the REFRESH SECURITY command with the option TYPE(SSL). This enables you to update some of your TLS settings without having to restart your channel initiator.

z/OS *Displaying security status*

To display the status of the security switches, and other security controls, issue the MQSC DISPLAY SECURITY command.

The following figure shows typical output of the DISPLAY SECURITY ALL command.

```
CSQH015I +CSQ1 Security timeout = 54 MINUTES
CSQH016I +CSQ1 Security interval = 12 MINUTES
CSQH030I +CSQ1 Security switches ...
CSQH034I +CSQ1 SUBSYSTEM: ON, 'SQ05.NO.SUBSYS.SECURITY' not found
CSQH032I +CSQ1 QMGR: ON, 'CSQ1.YES.QMGR.CHECKS' found
CSQH031I +CSQ1 QSG: OFF, 'SQ05.NO.QSG.CHECKS' found
CSQH031I +CSQ1 CONNECTION: OFF, 'CSQ1.NO.CONNECT.CHECKS' found
CSQH034I +CSQ1 COMMAND: ON, 'CSQ1.NO.COMMAND.CHECKS' not found
CSQH031I +CSQ1 CONTEXT: OFF, 'CSQ1.NO.CONTEXT.CHECKS' found
CSQH034I +CSQ1 ALTERNATE USER: ON, 'CSQ1.NO.ALTERNATE.USER.CHECKS' not found
CSQH034I +CSQ1 PROCESS: ON, 'CSQ1.NO.PROCESS.CHECKS' not found
CSQH034I +CSQ1 NAMELIST: ON, 'CSQ1.NO.NLIST.CHECKS' not found
CSQH034I +CSQ1 QUEUE: ON, 'CSQ1.NO.QUEUE.CHECKS' not found
CSQH034I +CSQ1 TOPIC: ON, 'CSQ1.NO.TOPIC.CHECKS' not found
CSQH031I +CSQ1 COMMAND RESOURCES: OFF, 'CSQ1.NO.CMD.RESC.CHECKS' found
CSQ9022I +CSQ1 CSQHPDTC ' DISPLAY SECURITY' NORMAL COMPLETION
```

Figure 18. Typical output from the DISPLAY SECURITY command

The example shows that the queue manager that replied to the command has subsystem, command, alternate user, process, namelist, and queue security active at queue manager level but not at queue sharing group level. Connection, command resource, and context security are not active. It also shows

that user ID timeouts are active, and that every 12 minutes the queue manager checks for user IDs that have not been used in this queue manager for 54 minutes and removes them.

Note: This command shows the current security status. It does not necessarily reflect the current status of the switch profiles defined to RACF, or the status of the RACF classes. For example, the switch profiles might have been changed since the last restart of this queue manager or REFRESH SECURITY command.

z/OS Security installation tasks for z/OS

After installing and customizing IBM MQ, authorize started task procedures to RACF, authorize access to various resources, and set up RACF definitions. Optionally, configure your system for TLS.

When IBM MQ is first installed and customized, you must perform these security-related tasks:

1. Set up IBM MQ data set and system security by:
   - Authorizing the queue manager started-task procedure xxxxMSTR and the distributed queuing started-task procedure xxxxCHIN to run under RACF.
   - Authorizing access to queue manager data sets.
   - Authorizing access to resources for those user IDs that will use the queue manager and utility programs.
   - Authorizing access for those queue managers that will use the coupling facility list structures.
   - Authorizing access for those queue managers that will use Db2.
2. Set up RACF definitions for IBM MQ security.
3. If you want to use Transport Layer Security (TLS), prepare your system to use certificates and keys.

z/OS *Setting up IBM MQ for z/OS data set security*

There are many types of IBM MQ user. Use RACF to control their access to system data sets.

The possible users of IBM MQ data sets include the following entities:

- The queue manager itself.
- The channel initiator
- IBM MQ administrators, who need to create IBM MQ data sets, run utility programs, and similar tasks.
- Application programmers who need to use the IBM MQ-supplied copybooks, include data sets, macros, and similar resources.
- Applications involving one or more of:
  - Batch jobs
  - TSO users
  - CICS regions
  - IMS regions
- Data sets CSQOUTX and CSQSNAP
- Dynamic queues SYSTEM.CSQXCMD.*

For all these potential users, protect the IBM MQ data sets with RACF.

You must also control access to all your 'CSQINP' data sets.

z/OS *RACF authorization of started-task procedures*

Some IBM MQ data sets are for the exclusive use of the queue manager. If you protect your IBM MQ data sets using RACF, you must also authorize the queue manager started-task procedure xxxxMSTR, and the distributed queuing started-task procedure xxxxCHIN, using RACF. To do this, use the STARTED class. Alternatively, you can use the started procedures table (ICHRIN03), but then you must perform an IPL of your z/OS system before the changes take effect.

For more information, see the *z/OS Security Server RACF System Programmer's Guide*.

The RACF user ID identified must have the required access to the data sets in the started-task procedure. For example, if you associate a queue manager started task procedure called CSQ1MSTR with the RACF user ID QMGRCSQ1, the user ID QMGRCSQ1 must have access to the z/OS resources accessed by the CSQ1 queue manager.

Also, the content of the GROUP field in the user ID of the queue manager must be the same as the content of the GROUP field in the STARTED profile for that queue manager. If the content in each GROUP field does not match then the appropriate user ID is prevented from entering the system. This situation causes IBM MQ to run with an undefined user ID and consequently close due to a security violation.

The RACF user IDs associated with the queue manager and channel initiator started task procedures must not have the TRUSTED attribute set.

z/OS *Authorizing access to data sets*

The IBM MQ data sets should be protected so that no unauthorized user can run a queue manager instance, or gain access to any queue manager data. To do this, use normal z/OS RACF data set protection.

Table 65 on page 258 summarizes the RACF access that the queue manager started task procedure must have to the different data sets.

Table 65. RACF access to data sets associated with a queue manager

RACF access	**Data sets**
READ	• `thlqual.SCSQAUTH` and `thlqual.SCSQANLx` (where x is the language letter for your national language). • The data sets referred to by CSQINP1, CSQINP2 and CSQXLIB in the queue manager's started task procedure. • SMDS data sets owned by other queue managers in the group. • Log, BSDS and archive log data sets for other queue managers in the group.
UPDATE	• All page sets and log and BSDS data sets. • SMDS data sets owned by a queue manager • SMDS data sets owned by other queue managers in the group, for the structures that the queue manager performs the RECOVER CFSTRUCT command.
ALTER	• All archive log data sets.

Table 66 on page 258 summarizes the RACF access that the started task procedure for distributed queuing must have to the different data sets.

Table 66. RACF access to data sets associated with distributed queuing

RACF access	**Data sets**
READ	• thlqual.SCSQAUTH, thlqual.SCSQANLx (where x is the language letter for your national language), and thlqual.SCSQMVR1. • LE library data sets. • The data sets referred to by CSQXLIB and CSQINPX in the channel initiator started task procedure.
UPDATE	• Data sets CSQOUTX and CSQSNAP

For more information, see the *z/OS Security Server RACF Security Administrator's Guide*.

z/OS *Encrypting data sets*

The IBM MQ data sets can be encrypted with z/OS data set encryption, so that the data is protected, or for regulatory reasons.

You can protect all page sets, active log, archive log, and bootstrap (BSDS) data sets with z/OS data set encryption.

Attention: You cannot protect shared message data sets (SMDS) with z/OS data set encryption by IBM MQ for z/OS 9.1.4 or earlier.

See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

z/OS *Setting up IBM MQ for z/OS resource security*

There are many types of IBM MQ user. Use RACF to control their access to IBM MQ resources.

The possible users of IBM MQ resources, such as queues and channels include the following entities:

- The queue manager itself.
- The channel initiator
- IBM MQ administrators, who need to create IBM MQ data sets, run utility programs, and similar tasks
- Application programmers who need to use the IBM MQ-supplied copybooks, include data sets, macros, and similar resources.
- Applications involving one or more of:
  - Batch jobs
  - TSO users
  - CICS regions
  - IMS regions
- Data sets CSQOUTX and CSQSNAP
- Dynamic queues SYSTEM.CSQXCMD.*

For all these potential users, protect the IBM MQ resources with RACF. In particular, note that the channel initiator needs access to various resources, as described in “Security considerations for the channel initiator on z/OS” on page 265, and so the user ID under which it runs must be authorized to access these resources.

If you are using a queue sharing group, the queue manager might issue various commands internally, so the user ID it uses must be authorized to issue such commands. The commands are:

- DEFINE, ALTER, and DELETE for every object that has QSGDISP(GROUP)
- START and STOP CHANNEL for every channel used with CHLDISP(SHARED)

z/OS *Configuring your z/OS system to use TLS*

Use this topic as example of how to configure IBM MQ for z/OS with Transport Layer Security (TLS) using RACF commands.

If you want to use TLS for channel security, there are a number of tasks you need to perform on your system. (For details on using RACF commands for certificates and key repositories (key rings), see Working with TLS on z/OS .)

1. Create a key ring in RACF to hold all the keys and certificates for your system, using the RACF RACDCERT command. For example:

```
RACDCERT ID(CHINUSER) ADDRING(QM1RING)
```

The ID must be either the channel initiator address space user ID or the user ID you want to own the key ring if it is to be a shared key ring.

2. Create a digital certificate for each queue manager, using the RACF RACDCERT command.

   The label of the certificate must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details. In this example it is `ibmWebSphereMQQM1`.

   For example:

```
RACDCERT ID(USERID) GENCERT
SUBJECTSDN(CN('username') O('IBM') OU('departmentname') C('England'))
WITHLABEL('ibmWebSphereMQQM1')
```

3. Connect the certificate in RACF to the key ring, using the RACF RACDCERT command. For example:

```
RACDCERT CONNECT(ID(USERID) LABEL('ibmWebSphereMQQM1') RING(QM1RING))
CONNECT ID(CHINUSER)
```

   You also need to connect any relevant signer certificates (from a certificate authority) to the key ring. That is, all certificate authorities for the TLS certificate of this queue manager and all certificate authorities for all TLS certificates that this queue manager communicates with. For example:

```
RACDCERT ID(CHINUSER)
CONNECT(CERTAUTH LABEL('My CA') RING(QM1RING) USAGE(CERTAUTH))
```

4. On each of your queue managers, use the IBM MQ ALTER QMGR command to specify the key repository that the queue manager needs to point to. For example, if the key ring is owned by the channel initiator address space:

```
ALTER QMGR SSLKEYR(QM1RING)
```

   or if you are using a shared key ring:

```
ALTER QMGR SSLKEYR(userid/QM1RING)
```

   where *userid* is the user ID that owns the shared key ring.

5. Certificate Revocation Lists (CRLs) allow the certificate authorities to revoke certificates that can no longer be trusted. CRLs are stored in LDAP servers. To access this list on the LDAP server, you first need to create an AUTHINFO object of AUTHTYPE CRLLDAP, using the IBM MQ DEFINE AUTHINFO command. For example:

```
DEFINE AUTHINFO(LDAP1)
AUTHTYPE(CRLLDAP)
CONNAME(ldap.server(389))
LDAPUSER('')
LDAPPWD('')
```

In this example, the certificate revocation list is stored in a public area of the LDAP server, so the LDAPUSER and LDAPPWD fields are not necessary.

Next, put your AUTHINFO object into a namelist, using the IBM MQ DEFINE NAMELIST command. For example:

```
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
```

Finally, associate the namelist with each queue manager, using the IBM MQ ALTER QMGR command. For example:

```
ALTER QMGR SSLCRLNL(LDAPNL)
```

6. Set up your queue manager to run TLS calls, using the IBM MQ ALTER QMGR command. This defines server subtasks that handle SSL calls only, which leaves the normal dispatchers to continue processing as normal without being affected by any SSL calls. You must have at least two of these subtasks. For example:

```
ALTER QMGR SSLTASKS(8)
```

   This change only takes effect when the channel initiator is restarted.

7. Specify the cipher specification to be used for each channel, using the IBM MQ DEFINE CHANNEL or ALTER CHANNEL command. For example:

```
ALTER CHANNEL(LDAPCHL)
CHLTYPE(SDR)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

   Both ends of the channel must specify the same cipher specification.

z/OS Managing channel authentication records in a QSG

Channel authentication records apply to the queue manager that they are created on, they are not shared throughout the queue sharing group (QSG). Therefore if all the queue managers in the queue sharing group are required to have the same rules, some management needs to be carried out to keep all the rules the consistent.

1. Always add the CMDSCOPE(*) option to all SET CHLAUTH commands. This will send the command to all running queue managers in the queue sharing group
2. Use the DISPLAY CHLAUTH command with the CMDSCOPE(*) option and then analyze the responses to see if the records are the same from all queue managers. When an inconsistency is found a SET CHLAUTH command can be issued containing the same rule with CMDSCOPE(*) or CMDSCOPE(*qmgr-name*).

3. Add a member to the queue manager's CSQINP2 concatenation (see Initialization commands for details) that has the full set of rules. These will be read as part of the queue manager's initialization process. If the `SET CHLAUTH` command uses `ACTION(ADD)` the rule will only be added if it didn't exist. Using `ACTION(REPLACE)` will replace an existing rule if it already exists or add it if it does not. The same member could then be placed in the CSQINP2 concatenation of all queue managers in the queue sharing group.
4. Use the CSQUTIL utility (see Issuing commands to IBM MQ (COMMAND) for details) to extract the rules from one queue manager using either the `MAKEDEF` or `MAKEREP` option. Then replay the output using CSQUTIL into the target queue manager.

Related concepts

Channel authentication records

Para exercer um controle mais preciso sobre o acesso concedido à conexão de sistemas em um nível de canal, é possível usar registros de autenticação de canal.

z/OS Auditing considerations on z/OS

The normal RACF auditing controls are available for conducting a security audit of a queue manager. IBM MQ does not gather any security statistics of its own. The only statistics are those that can be created by auditing.

RACF auditing can be based upon:

- User IDs
- Resource classes
- Profiles

For more details, see the *z/OS Security Server RACF Auditor's Guide*.

Note: Auditing degrades performance; the more auditing you implement, the more performance is degraded. This is also a consideration for the use of the RACF WARNING option.

z/OS *Auditing RESLEVEL*

Use the RESAUDIT system parameter to control the production of RESLEVEL audit records. RACF GENERAL audit records are produced.

Produce RESLEVEL audit records by setting the RESAUDIT system parameter to YES. If the RESAUDIT parameter is set to NO, audit records are not produced. For more details about setting this parameter, see Using CSQ6SYSP.

If RESAUDIT is set to YES, no normal RACF audit records are taken when the RESLEVEL check is made to see what access an address space user ID has to the hlq.RESLEVEL profile. Instead, IBM MQ requests that RACF create a GENERAL audit record (event number 27). These checks are only carried out at connect time, so the performance cost is minimal.

Attention: RACFRW is no longer the suggested utility for processing RACF audit records. You should use the RACF SMF data unload utility as this is the preferred reporting method.

You can report the IBM MQ general audit records using the RACF report writer (RACFRW). You could use the following RACFRW commands to report the RESLEVEL access:

```
RACFRW
SELECT PROCESS
EVENT GENERAL
LIST
END
```

A sample report from RACFRW, excluding the *Date*, *Time*, and *SYSID* fields, is shown in Figure 19 on page 263.

```
      RACF REPORT - LISTING OF PROCESS RECORDS                                  PAGE   4
                                E
                                V  Q
                                E  U
*JOB/USER *STEP/  --TERMINAL--  N  A
   NAME    GROUP     ID    LVL  T  L

 WS21B    MQMGRP IGJZM000   0   27 0  JOBID=(WS21B 05.111 09:44:57),USERDATA=()
    TRUSTED USER                      AUTH=(NONE),REASON=(NONE)
                                SESSION=TSOLOGON,TERMINAL=IGJZM000,
                                LOGSTR='CSQH RESLEVEL CHECK PERFORMED AGAINST
PROFILE(QM66.RESLEVEL),
                                CLASS(MQADMIN), ACCESS EQUATES TO
(CONTROL)',RESULT=SUCCESS,MQADMIN
```

Figure 19. Sample output from RACFRW showing RESLEVEL general audit records

From checking the LOGSTR data in this sample output, you can see that TSO user WS21B has CONTROL access to QM66.RESLEVEL. This means that all resource security checks are bypassed when user WS21B access QM66 resources.

For more information about using RACFRW, see The RACF report writer in the *z/OS Security Server RACF Auditor's Guide*.

z/OS Customizing security

If you want to change the way IBM MQ security operates, you must do this through the SAF exit (ICHRFR00), or exits in your external security manager.

To find out more about RACF exits, see the *z/OS Security Server RACROUTE Macro Reference* documentation.

Note: Because IBM MQ optimizes calls to the ESM, RACROUTE requests might not be made on, for example, every open for a particular queue by a particular user.

z/OS Security violation messages on z/OS

A security violation is indicated by the return code MQRC_NOT_AUTHORIZED in an application program or by a message in the job log.

A return code of MQRC_NOT_AUTHORIZED can be returned to an application program for the following reasons:

- A user is not allowed to connect to the queue manager. In this case, you get an ICH408I message in the Batch/TSO, CICS, or IMS job log.
- A user sign-on to the queue manager has failed because, for example, the job user ID is not valid or appropriate, or the task user ID or alternate user ID is not valid. One or more of these user IDs might not be valid because they have been revoked or deleted. In this case, you get an ICHxxxx message and possibly an IRRxxxx message in the queue manager job log giving the reason for the sign-on failure. For example:

```
ICH408I USER(NOTDFND ) GROUP(        ) NAME(???                 )
  LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND
```

- An alternate user has been requested, but the job or task user ID does not have access to the alternate user ID. For this failure, you get a violation message in the job log of the relevant queue manager.
- A context option has been used or is implied by opening a transmission queue for output, but the job user ID or, where applicable, the task or alternate user ID does not have access to the context option. In this case, a violation message is put in the job log of the relevant queue manager.

- An unauthorized user has attempted to access a secured queue manager object, for example, a queue. In this case, an ICH408I message for the violation is put in the job log of the relevant queue manager. This violation might be due to the job or, when applicable, the task or alternate user ID.

Violation messages for command security and command resource security can also be found in the job log of the queue manager.

If the ICH408I violation message shows the queue manager jobname rather than a user ID, this is normally the result of a blank alternate user ID being specified. For example:

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          INSUFFICIENT ACCESS AUTHORITY
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

You can find out who is allowed to use blank alternate user IDs by checking the access list of the MQADMIN profile hlq.ALTERNATE.USER.-BLANK-.

An ICH408I violation message can also be generated by:

- A command being sent to the system-command input queue without context. User-written programs that write to the system-command input queue should always use a context option. For more information, see “Profiles for context security” on page 223.
- When the job accessing the IBM MQ resource does not have a user ID associated with it, or when an IBM MQ adapter cannot extract the user ID from the adapter environment.

Violation messages might also be issued if you are using both queue sharing group and queue manager level security. You might get messages indicating that no profile has been found at queue manager level, but still be granted access because of a queue sharing group level profile.

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

See the z/OS for Security Server RACF Messages and Codes documentation for more information on ICH408I messages.

z/OS What to do if access is allowed or disallowed incorrectly

In addition to the information detailed in the z/OS documentation, use this checklist if access to a resource appears to be incorrectly controlled.

See the *z/OS Security Server RACF Security Administrator's Guide* for the detailed steps if access is allowed or disallowed.

- Are the switch profiles correctly set?
  - Is RACF active?
  - Are the IBM MQ RACF classes installed and active?

    Use the RACF command, SETROPTS LIST, to check this.
  - Use the IBM MQ DISPLAY SECURITY command to display the current switch status from the queue manager.
  - Check the switch profiles in the MQADMIN class.

    Use the RACF commands, SEARCH and RLIST, for this.
  - Recheck the RACF switch profiles by issuing the IBM MQ REFRESH SECURITY(MQADMIN) command.

- Has the RACF resource profile changed? For example, has universal access on the profile changed or has the access list of the profile changed?
  - Is the profile generic?

    If it is, issue the RACF command, SETROPTS GENERIC(classname) REFRESH.
  - Have you refreshed the security on this queue manager?

    If required, issue the RACF command SETROPTS RACLIST(classname) REFRESH.

    If required, issue the IBM MQ REFRESH SECURITY(*) command.
- Has the RACF definition of the user changed? For example, has the user been connected to a new group or has the user access authority been revoked?
  - Have you reverified the user by issuing the IBM MQ RVERIFY SECURITY(userid) command?
- Are security checks being bypassed due to RESLEVEL?
  - Check the connecting user ID's access to the RESLEVEL profile. Use the RACF audit records to determine what the RESLEVEL is set to.
  - For channels, remember that the access level that the channel initiator's userid has to RESLEVEL is inherited by all channels, so an access level, such as ALTER, that causes all checks to be bypassed causes security checks to be bypassed for all channels.
  - If you are running from CICS, check the transaction's RESSEC setting.
  - If RESLEVEL has been changed while a user is connected, they must disconnect and reconnect before the new RESLEVEL setting takes effect.
- Are you using queue sharing groups?
  - If you are using both queue sharing group and queue manager level security, check that you have defined all the correct profiles. If queue manager profile is not defined, a message is sent to the log stating that the profile was not found.
  - Have you used a combination of switch settings that is not valid so that full security checking has been set on?
  - Do you need to define security switches to override some of the queue sharing group settings for your queue manager?
  - Is a queue manager level profile taking precedence over a queue sharing group level profile?

z/OS Security considerations for the channel initiator on z/OS

If you are using resource security in a distributed queuing environment, the Channel initiator address space needs appropriate access to various IBM MQ resources. You can use the Integrated Cryptographic Support Facility (ICSF) to seed the password protection algorithm.

See the z/OS Cryptographic Services documentation for more information on ICSF.

Using resource security

If you are using resource security, consider the following points if you are using distributed queuing:

System queues
The channel initiator address space needs RACF UPDATE access to the system queues listed at “System queue security” on page 213, and to all the user destination queues and the dead-letter queue (but see “Dead-letter queue security” on page 212).

Transmission queues
The channel initiator address space needs ALTER access to all the user transmission queues.

Context security
The channel user ID (and the MCA user ID if one has been specified) need RACF CONTROL access to the hlq.CONTEXT.queuename profiles in the MQADMIN class. Depending on the RESLEVEL profile, the channel user ID might also need CONTROL access to these profiles.

All channels need CONTROL access to the MQADMIN hlq.CONTEXT. dead-letter-queue profile. All channels (whether initiating or responding) can generate reports, and consequently they need CONTROL access to the hlq.CONTEXT.reply-q profile.

SENDER, CLUSSDR, and SERVER channels need CONTROL access to the hlq.CONTEXT.xmit-queue-name profiles since messages can be put onto the transmission queue to wake up the channel to end gracefully.

Note: If the channel user ID, or a RACF group to which the channel user ID is connected, has CONTROL or ALTER access to the hlq.RESLEVEL, then there are no resource checks for the channel initiator or any of its channels.

See "Profiles for context security" on page 223 "RESLEVEL and the channel initiator connection" on page 241 and "User IDs for security checking on z/OS" on page 243 for more information.

CSQINPX
If you are using the CSQINPX input data set, the channel initiator also needs READ access to CSQINPX, and UPDATE access to data set CSQOUTX and dynamic queues SYSTEM.CSQXCMD.*.

Connection security
The channel initiator address space connection requests use a connection type of CHIN, for which appropriate access security must be set, see "Connection security profiles for the channel initiator" on page 206.

Data sets
The channel initiator address space needs appropriate access to queue manager data sets, see "Authorizing access to data sets" on page 258.

Commands
The distributed queuing commands (for example, DEFINE CHANNEL, START CHINIT, START LISTENER, and other channel commands) must have appropriate command security set, see Table 49 on page 226.

If you are using a queue sharing group, the channel initiator might issue various commands internally, so the user ID it uses must be authorized to issue such commands. These commands are START and STOP CHANNEL for every channel used with CHLDISP(SHARED).

If the PSMODE of the queue manager is not DISABLED, the channel initiator must have READ access to the DISPLAY PUBSUB command.

Channel security
Channels, particularly receivers and server-connections, need appropriate security to be set up; see "User IDs for security checking on z/OS" on page 243 for more information.

You can also use the Transport Layer Security (TLS) protocol to provide security on channels. See "Protocolos de segurança TLS no IBM MQ" on page 25 for more information about using TLS with IBM MQ.

See also "Controle de acesso para clientes" on page 107 for information about server-connection security.

User IDs
The user IDs described in "User IDs used by the channel initiator" on page 246 and "User IDs used by the intra-group queuing agent" on page 250 need the following access:

- RACF UPDATE access to the appropriate destination queues and the dead-letter queue
- RACF CONTROL access to the `hlq.CONTEXT.queuename` profile if context checking is performed at the receiver
- Appropriate access to the hlq.ALTERNATE.USER.userid profiles they might need to use.
- For clients, the appropriate RACF access to the resources to be used.

APPC security
Set appropriate APPC security if you are using the LU 6.2 transmission protocol. (Use the APPCLU RACF class for example.) For information about setting up security for APPC, see the following documentation:

- *z/OS MVS Planning: APPC Management*
- *z/OS MVS Programming: Writing Servers for APPC/MVS*

Outbound transmissions use the "SECURITY(SAME)" APPC option. As a result, the user ID of the channel initiator address space and its default profile (RACF GROUP) are flowed across the network to the receiver with an indicator that the user ID has already been verified (ALREADYV).

If the receiving side is also z/OS, the user ID and profile are verified by APPC and the user ID is presented to the receiver channel and used as the channel user ID.

In an environment where the queue manager is using APPC to communicate with another queue manager on the same or another z/OS system, you need to ensure that either:

- The VTAM definition for the communicating LU specifies SETACPT(ALREADYV)
- There is a RACF APPCLU profile for the connection between LUs that specifies CONVSEC(ALREADYV)

Changing security settings
If the RACF access level that either the channel user ID or MCA user ID has to a destination queue is changed, this change takes effect only for new object handles (that is, new MQOPEN s) for the destination queue. The times when MCAs open and close queues is variable; if a channel is already running when such an access change is made, the MCA can continue to put messages on the destination queue using the existing security access of the user IDs rather than the updated security access. Stopping and restarting the channels to enforce the updated access level avoids this scenario.

Automatic restart
If you are using the z/OS Automatic Restart Manager (ARM) to restart the channel initiator, the user ID associated with the XCFAS address space must be authorized to issue the IBM MQ START CHINIT command.

Using the Integrated Cryptographic Service Facility (ICSF)

The channel initiator can use ICSF to generate a random number when seeding the password protection algorithm to obfuscate passwords flowing over client channels if TLS is not being used. The process of generating a random number is called *entropy*.

If you have the z/OS feature installed but have not started ICSF, you see message CSQX213E and the channel initiator uses STCK for entropy.

Message CSQX213E warns you that the password protection algorithm is not as secure as it could be. However, you can continue your process; there is no other impact on runtime.

If you do not have the z/OS feature installed, the channel initiator automatically uses STCK.

Notes:

1. Using ICSF for entropy generates more random sequences than using STCK.
2. If you start ICSF you must restart the channel initiator.
3. ICSF is required for certain CipherSpecs. If you attempt to use one of these CipherSpecs and you do not have ICSF installed, you receive message CSQX629E.

z/OS Security in queue manager clusters on z/OS

Security considerations for clusters are the same for queue managers and channels that are not clustered. The channel initiator needs access to some additional system queues, and some additional commands need appropriate security set.

You can use the MCA user ID, channel authentication records, TLS, and security exits to authenticate cluster channels (as with conventional channels). The channel authentication records or security exit relating to the cluster-receiver channel must check that the remote queue manager is permitted access to the server queue manager's cluster queues. You can start to use IBM MQ cluster support without

changing your existing queue access security. You must, however, allow other queue managers in the cluster to write to the SYSTEM.CLUSTER.COMMAND.QUEUE if they are to join the cluster.

IBM MQ cluster support does not provide a mechanism to limit a member of a cluster to the client role only. As a result, you must be sure that you trust any queue managers that you allow into the cluster. If any queue manager in the cluster creates a queue with a particular name, it can receive messages for that queue, regardless of whether the application putting messages to that queue intended this or not.

To restrict the membership of a cluster, take the same action that you would take to prevent queue managers connecting to receiver channels. You restrict the membership of a cluster by using channel authentication records or by writing a security exit program on the receiver channel. You can also write an exit program to prevent unauthorized queue managers from writing to the SYSTEM.CLUSTER.COMMAND.QUEUE.

Note: It is not advisable to permit applications to open the SYSTEM.CLUSTER.TRANSMIT.QUEUE directly. It is also not advisable to permit an application to open any other transmission queue directly.

If you are using resource security, consider the following points in addition to the considerations contained in "Security considerations for the channel initiator on z/OS" on page 265:

System queues
The channel initiator needs RACF ALTER access to the following system queues:

- SYSTEM.CLUSTER.COMMAND QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE.

and UPDATE access to SYSTEM.CLUSTER.REPOSITORY.QUEUE

It also needs READ access to any namelists used for clustering.

Commands
Set appropriate command security (as described in Table 49 on page 226) for the cluster support commands (REFRESH and RESET CLUSTER, SUSPEND, and RESUME QMGR.

z/OS Security considerations for using IBM MQ with CICS

All the CICS versions supported by IBM MQ 9.0.0, and later, use the CICS supplied version of the adapter and bridge.

For details of security considerations, see:

- Security for the CICS-MQ adapter.
- Security for the CICS-MQ bridge.

z/OS Security considerations for using IBM MQ with IMS

Use this topic to plan your security requirements when you use IBM MQ with IMS.

Using the OPERCMDS class

If you are using RACF to protect resources in the OPERCMDS class, ensure that the userid associated with your IBM MQ queue manager address space has authority to issue the MODIFY command to any IMS system to which it can connect.

Security considerations for the IMS bridge

There are four aspects that you must consider when deciding your security requirements for the IMS bridge, these are:

- What security authorization is needed to connect IBM MQ to IMS
- How much security checking is performed on applications using the bridge to access IMS
- Which IMS resources these applications are allowed to use

- What authority is to be used for messages that are put and got by the bridge

When you define your security requirements for the IMS bridge you must consider the following:

- Messages passing across the bridge might have originated from applications on platforms that do not offer strong security features
- Messages passing across the bridge might have originated from applications that are not controlled by the same enterprise or organization

z/OS *Security considerations for connecting to IMS*

Grant the user ID of the IBM MQ queue manager address space access to the OTMA group.

The IMS bridge is an OTMA client. The connection to IMS operates under the user ID of the IBM MQ queue manager address space. This is normally defined as a member of the started task group. This user ID must be granted access to the OTMA group (unless the /SECURE OTMA setting is NONE).

To do this, define the following profile in the FACILITY class:

```
IMSXCF.xcfgname.mqxcfmname
```

Where `xcfgname` is the XCF group name and `mqxcfmname` is the XCF member name of IBM MQ.

You must give your IBM MQ queue manager user ID read access to this profile.

Note:

1. If you change the authorities in the FACILITY class, you must issue the RACF command SETROPTS RACLIST(FACILITY) REFRESH to activate the changes.
2. If profile hlq.NO.SUBSYS.SECURITY exists in the MQADMIN class, no user ID is passed to IMS and the connection fails unless the /SECURE OTMA setting is NONE.

z/OS *Application access control for the IMS bridge*

Define a RACF profile in the FACILITY class for each IMS system. Grant an appropriate level of access to the IBM MQ queue manager user ID.

For each IMS system that the IMS bridge connects to, you can define the following RACF profile in the FACILITY class to determine how much security checking is performed for each message passed to the IMS system.

```
IMSXCF.xcfgname.imsxcfmname
```

Where `xcfgname` is the XCF group name and `imsxcfmname` is the XCF member name for IMS. (You need to define a separate profile for each IMS system.)

The access level you allow for the IBM MQ queue manager user ID in this profile is returned to IBM MQ when the IMS bridge connects to IMS, and indicates the level of security that is required on subsequent transactions. For subsequent transactions, IBM MQ requests the appropriate services from RACF and, where the user ID is authorized, passes the message to IMS.

OTMA does not support the IMS /SIGN command; however, IBM MQ allows you to set the access checking for each message to enable implementation of the necessary level of control.

The following access level information can be returned:

NONE or NO PROFILE FOUND
: These values indicate that maximum security is required, that is, authentication is required for every transaction. A check is made to verify that the user ID specified in the *UserIdentifier* field of the MQMD structure, and the password or PassTicket in the *Authenticator* field of the MQIIH structure are known

to RACF, and are a valid combination. A UTOKEN is created with a password or PassTicket, and passed to IMS ; the UTOKEN is not cached.

Note: If profile hlq.NO.SUBSYS.SECURITY exists in the MQADMIN class, this level of security overrides whatever is defined in the profile.

READ
This value indicates that the same authentication is to be performed as for NONE under the following circumstances:

- The first time that a specific user ID is encountered
- When the user ID has been encountered before but the cached UTOKEN was not created with a password or PassTicket

IBM MQ requests a UTOKEN if required, and passes it to IMS.

Note: If a request to reverify security has been acted on, all cached information is lost and a UTOKEN is requested the first time each user ID is later encountered.

UPDATE
A check is made that the user ID in the *UserIdentifier* field of the MQMD structure is known to RACF.

A UTOKEN is built and passed to IMS ; the UTOKEN is cached.

CONTROL/ALTER
These values indicate that no security UTOKENs need to be provided for any user IDs for this IMS system. (You would probably only use this option for development and test systems.)

Attention: Note that the user ID contained in the *UserIdentifier* field of the MQMD structure is still passed for **CONTROL/ALTER**.

Note:

1. This access is defined when IBM MQ connects to IMS, and lasts for the duration of the connection. To change the security level, the access to the security profile must be changed and then the bridge stopped and restarted (for example, by stopping and restarting OTMA).
2. If you change the authorities in the FACILITY class, you must issue the RACF command SETROPTS RACLIST(FACILITY) REFRESH to activate the changes.
3. You can use a password or a PassTicket, but you must remember that the IMS bridge does not encrypt data. For information about using PassTickets, see “Using RACF PassTickets in the IMS header” on page 271.
4. Some of these results might be affected by security settings in IMS, using the /SECURE OTMA command.
5. Cached UTOKEN information is held for the duration defined by the INTERVAL and TIMEOUT parameters of the IBM MQ ALTER SECURITY command.
6. The RACF WARNING option has no effect on the IMSXCF.xcfgname.imsxcfmname profile. Its use does not affect the level of access granted, and no RACF WARNING messages are produced.

z/OS *Security checking on IMS*

Messages that pass across the bridge contain security information. The security checks made depend on the setting of the IMS command /SECURE OTMA.

Each IBM MQ message that passes across the bridge contains the following security information:

- A user ID contained in the *UserIdentifier* field of the MQMD structure
- The security scope contained in the *SecurityScope* field of the MQIIH structure (if the MQIIH structure is present)
- A UTOKEN (unless the IBM MQ sub system has CONTROL or ALTER access to the relevant `IMSXCF.xcfgname.imsxcfmname` profile)

The security checks made depend on the setting of the IMS command /SECURE OTMA, as follows:

/SECURE OTMA NONE
No security checks are made for the transaction.

/SECURE OTMA CHECK
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking.

An ACEE (Accessor Environment Element) is built in the IMS control region.

/SECURE OTMA FULL
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking.

An ACEE is built in the IMS dependent region as well as the IMS control region.

/SECURE OTMA PROFILE
The *UserIdentifier* field of the MQMD structure is passed to IMS for transaction or command authority checking

The *SecurityScope* field in the MQIIH structure is used to determine whether to build an ACEE in the IMS dependent region as well as the control region.

Note:

1. If you change the authorities in the TIMS or CIMS class, or the associated group classes GIMS or DIMS, you must issue the following IMS commands to activate the changes:
   - /MODIFY PREPARE RACF
   - /MODIFY COMMIT
2. If you do not use /SECURE OTMA PROFILE, any value specified in the **SecurityScope** field of the MQIIH structure is ignored.

z/OS *Security checking done by the IMS bridge*

Different authorities are used depending on the action being performed.

When the bridge puts or gets a message, the following authorities are used:

Getting a message from the bridge queue
No security checks are performed.

Putting an exception, or COA report message
Uses the authority of the user ID in the *UserIdentifier* field of the MQMD structure.

Putting a reply message
Uses the authority of the user ID in the *UserIdentifier* field of the MQMD structure of the original message

Putting a message to the dead-letter queue
No security checks are performed.

Note:

1. If you change the IBM MQ class profiles, you must issue the IBM MQ REFRESH SECURITY(*) command to activate the changes.
2. If you change the authority of a user, you must issue the MQSC RVERIFY SECURITY command to activate the change.

z/OS *Using RACF PassTickets in the IMS header*

You can use a PassTicket in place of a password in the IMS header.

If you want to use a PassTicket instead of a password in the IMS header (MQIIH), specify the application name against which the PassTicket is validated in the PASSTKTA attribute of the STGCLASS definition of the IMS bridge queue to which the message is to be routed.

If the PASSTKTA value is left blank, you must arrange to have a PassTicket generated. The application name in this case must be of the form MVSxxxx, where xxxx is the SMFID of the z/OS system on which the target queue manager runs.

A PassTicket is built from a user ID, the target application name, and a secret key. It is an 8-byte value containing uppercase alphabetic and numeric characters. It can be used only once, and is valid for a 20 minute period. If a PassTicket is generated by a local RACF system, RACF only checks that the profile exists and not that the user has authority against the profile. If the PassTicket was generated on a remote system, RACF validates the access of the user ID to the profile. For full information about PassTickets, see the *z/OS Security Server RACF Security Administrator's Guide*.

PassTickets in IMS headers are given to RACF by IBM MQ, not IMS.

z/OS Migrating a z/OS queue manager to mixed-case security

Follow these steps to migrate a queue manager to mixed-case security. You review the level of security product you are using and activate the new IBM MQ external security manager classes. Run the **REFRESH SECURITY** command to activate the mixed-case profiles.

Before you begin

1. Ensure all IBM MQ external security manager classes are activated.
2. Ensure your queue manager is started.

About this task

Follow these steps to convert a queue manager to mixed-case security.

Procedure

1. Copy all your existing profiles and access levels from the uppercase classes to the equivalent mixed-case external security manager class.
   a) `MQADMIN` to `MXADMIN`.
   b) `MQPROC` to `MXPROC`.
   c) `MQNLIST` to `MXNLIST`.
   d) `MQQUEUE` to `MXQUEUE`.
2. Change the value of the `SCYCASE` queue manager attribute to `MIXED` by issuing the following command.

```
ALTER QMGR SCYCASE(MIXED)
```

3. Activate the security profiles by issuing the following command.

```
REFRESH SECURITY(*) TYPE(CLASSES)
```

4. Test that your security profiles are working correctly.

What to do next

Review your object definitions and create new mixed-case profiles as appropriate, using the **REFRESH SECURITY** command as required to activate the profiles.

Configurando a Segurança do IBM MQ MQI client

Deve-se considerar a segurança do IBM MQ MQI client, para que os aplicativos clientes não tenham acesso sem restrição aos recursos no servidor.

Ao executar um aplicativo cliente, não execute o aplicativo usando um ID do usuário que tenha mais direitos de acesso do que necessário; por exemplo, um usuário no grupo `mqm` ou até mesmo o usuário `mqm` em si.

Ao executar um aplicativo como um usuário com direitos de acesso em excesso, você corre o risco de o aplicativo acessar e alterar as partes do gerenciador de filas, seja por acaso ou intencionalmente.

Existem dois aspectos de segurança entre um aplicativo cliente e seu servidor do gerenciador de filas: autenticação e controle de acesso.

- A autenticação pode ser usada para assegurar que o aplicativo cliente, em execução como um usuário específico, é quem eles dizem que são. Ao usar autenticação é possível evitar que um invasor ganhe acesso ao seu gerenciador de filas personificando um de seus aplicativos.

  A autenticação é fornecida por uma de duas opções:

  – O recurso de autenticação de conexão.

    Para obter mais informações sobre autenticação de conexão, consulte “Autenticação de conexão” na página 74.

  – Usando autenticação mútua dentro do TLS.

    Para obter informações adicionais sobre TLS, veja “Trabalhando com SSL/TLS” na página 280.

- O controle de acesso pode ser usado para conceder ou remover direitos de acesso para um usuário ou grupo específico de usuários. Ao executar um aplicativo cliente com um usuário especificamente criado (ou usuário em um grupo específico), é possível usar controles de acesso para assegurar que o aplicativo não possa acessar partes de seu gerenciador de filas que ele não deve.

  Ao configurar o controle de acesso, deve-se considerar as regras de autenticação de canal e o campo MCAUSER em um canal. Ambos os recursos têm a capacidade de mudar o ID do usuário que está sendo usado para verificar os direitos de controle de acesso.

  Para obter informações adicionais sobre controle de acesso, consulte o “Autorizando o acesso aos objetos” na página 357.

Se você configurou um aplicativo cliente para se conectar a um canal específico com um ID restrito, mas o canal tem um ID de administrador configurado em seu campo MCAUSER, então, considerando que o aplicativo cliente se conecte com sucesso, o ID de administrador é usado para verificações do controle de acesso. Portanto, o aplicativo cliente terá direitos de acesso completos ao seu gerenciador de filas.

Para obter mais informações sobre o atributo MCAUSER, consulte “Mapeando um ID de usuário cliente para um ID do usuário MCAUSER” na página 393.

As regras de autenticação de canal também podem ser usadas como um método para controlar o acesso a um gerenciador de filas, configurando regras e os critérios específicos para uma conexão para ser aceito.

Para obter mais informações sobre as regras de autenticação de canal, consulte: “Registros de Autenticação de Canal” na página 53.

Especificando que Apenas CipherSpecs Certificados por FIPS São Usados no Tempo de Execução no Cliente de MQI

Crie seus repositórios de chaves usando software compatível com FIPS e, em seguida, especifique que o canal deve usar CipherSpecs certificados por FIPS.

Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o IBM Crypto for C (ICC) certificado e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na NIST CMVP modules in process list.

O IBM MQ Operator 3.2.0 e a imagem do contêiner do gerenciador de filas 9.4.0.0 em diante são baseados no UBI 9 A conformidade do FIPS 140-3 está pendente atualmente e seu status pode ser

visualizado procurando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" no NIST CMVP modules in process list.

Para ser compatível com FIPS no tempo de execução, os repositórios de chaves devem ter sido criados e gerenciados usando apenas software compatível com FIPS, como **runmqakm** com a opção `-fips` .

É possível especificar que um canal TLS deve usar somente CipherSpecs certificados por FIPS de três maneiras, listadas por ordem de precedência:

1. Configure o campo FipsRequired na estrutura do MQSCO como `MQSSL_FIPS_YES`
2. Configure a variável de ambiente **MQSSLFIPS** como YES.
3. Configure o atributo **SSLFipsRequired** na sub-rotina SSL do arquivo de configuração do cliente como YES

Por padrão, CipherSpecs certificados por FIPS não são obrigatórios.

Esses valores têm os mesmos significados que os valores de parâmetro equivalentes em **ALTER QMGR SSLFIPS** (consulte **ALTER QMGR** (alterar configurações do gerenciador de filas)). Se atualmente o processo do cliente não tiver conexões TLS ativas e um valor FipsRequired for especificado validamente em SSL MQCONNX, todas as conexões TLS subsequentes associadas a esse processo deverão usar somente CipherSpecs associados a esse valor. Isso se aplica até que esta e todas as outras conexões TLS sejam interrompidas, em cujo estágio um MQCONNX subsequente pode fornecer um novo valor para FipsRequired.

Se o hardware de criptografia estiver presente, os módulos de criptografia usados pelo IBM MQ poderão ser configurados para ser aqueles módulos fornecidos pelo produto de hardware e eles poderão ser certificados por FIPS em um nível específico. Os módulos configuráveis, e se eles são certificados por FIPS, dependem do produto de hardware em uso.

Quando possível, se CipherSpecs somente FIPS for configurado, o cliente MQI rejeitará conexões que especificam um CipherSpec não FIPS com `MQRC_SSL_INITIALIZATION_ERROR`. O IBM MQ não garante rejeitar todas essas conexões e é sua responsabilidade determinar se sua configuração do IBM MQ está com o padrão FIPS.

Conceitos relacionados

“Federal Information Processing Standards (FIPS) para AIX, Linux, and Windows” na página 36
Quando a criptografia é necessária em um canal SSL/TLS em AIX, Linux, and Windows sistemas, IBM MQ usa um pacote de criptografia chamado IBM Crypto for C (ICC). Nas plataformas do AIX, Linux, and Windows , o software ICC passou no Programa de Validação de Criptomódulos do Federal Information Processing Standards (FIPS) do US National Institute of Standards and Technology, no nível 140-2.

AIX Executando aplicativos clientes TLS com várias instalações do GSKit 8.0 on AIX .

Os aplicativos clientes TLS no AIX podem ter `MQRC_CHANNEL_CONFIG_ERROR` e erro `AMQ6175` ao executar em sistemas AIX com diversas instalações do IBM Global Security Kit (GSKit) 8.0

Ao executar aplicativos clientes em um sistema AIX com várias instalações do GSKit 8.0 , as chamadas de conexão do cliente podem retornar `MQRC_CHANNEL_CONFIG_ERROR` ao usar TLS Os logs do `/var/mqm/errors` registram os erros `AMQ6175` e `AMQ9220` para o aplicativo cliente com falha, por exemplo:

```
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 ----------------------------------------------------------
```

Uma causa comum desse erro é que a configuração da variável de ambiente LIBPATH ou LD_LIBRARY_PATH fez com que o cliente IBM MQ carregasse um conjunto combinado de bibliotecas de duas instalações diferentes do GSKit 8.0 . Executar um aplicativo cliente do IBM MQ em um ambiente do Db2 pode causar este erro.

Para evitar este erro, inclua os diretórios da biblioteca do IBM MQ na frente do caminho da biblioteca para que as bibliotecas do IBM MQ tenham precedência. Isso pode ser alcançado usando o comando **setmqenv** com o parâmetro **-k**, por exemplo:

```
. /usr/mqm/bin/setmqenv -s -k
```

Para obter informações adicionais sobre o uso do comando **setmqenv**, consulte o setmqenv (configurar o ambiente IBM MQ)

Configurando canais TLS com MQSC

Para configurar canais TLS, use os comandos **runmqsc** e ALTER CHANNEL. É possível opcionalmente configurar um canal para aceitar somente certificados com atributos no nome distinto do proprietário que corresponde a valores fornecidos. Também é possível opcionalmente configurar um canal do gerenciador de filas para que o gerenciador de filas recuse a conexão se a parte iniciante não enviar seu próprio certificado pessoal.

Sobre esta tarefa

Para configurar canais no IBM MQ Explorer, consulte Configurando canais TLS com o IBM MQ Explorer.

Para configurar canais usando **runmqsc**, conclua as etapas a seguir.

Procedimento

1. Chame o comando **runmqsc** conectando ao gerenciador de filas de destino.
2. Identifique o canal que deseja ativar para TLS.

   Observe o nome do canal e o tipo de canal.
3. Use o comando ALTER CHANNEL para alterar várias propriedades de um canal IBM MQ .

   Você fornece o nome e o tipo de canal, além do comando. Por exemplo, alterar um canal emissor chamado MQ.TEST execute o comando a seguir:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR)
```

Há vários atributos de canal relacionados ao TLS que podem ser ajustadas nas definições de canais do IBM MQ

Como proceder a seguir

Configurando a Segurança das Mensagens

O sistema de mensagens ativado para TLS oferece dois métodos para assegurar a segurança da mensagem:

- A criptografia assegura que se a mensagem for interceptada, será ilegível.
- As funções hash asseguram que se a mensagem for alterada, isso será detectado.

A combinação desses métodos é chamada de especificação de criptografia ou CipherSpec. O mesmo CipherSpec deve ser configurado para as duas extremidades de um canal, caso contrário, o sistema de mensagens ativado para TLS falhará. Para obter informações adicionais, consulte "Segurança do IBM MQ" na página 7.

Para alterar um canal do IBM MQ ativar TLS, especifique um valor no atributo SSLCIPH. Este atributo deve ser configurado como um CipherSpec válido para a plataforma de fila do gerenciador de filas da lista "Ativando CipherSpecs" na página 427

Para alterar um canal IBM MQ para desativar TLS, configure SSLCIPH para um valor em branco. Por exemplo:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCIPH(ANY_TLS12_OR_HIGHER)
```

Nota: Você deve colocar o nome do canal entre aspas simples para assegurar que as maiúsculas e minúsculas do caractere sejam mantidas. Sem aspas simples, o IBM MQ transforma a sequência em maiúsculas.

Filtrando Certificados pelo Nome do Proprietário

Os certificados contêm o nome distinto do proprietário do certificado. Opcionalmente, é possível configurar o canal para aceitar apenas certificados com atributos no nome distinto do proprietário que correspondam a valores fornecidos.

Os nomes de atributos que o IBM MQ pode filtrar são listados na tabela a seguir:

Nomes de atributos	**Significado**
SERIALNUMBER	Número de série do certificado
MAIL	Endereço de e-mail
Deprecated E	Endereço de e-mail (descontinuado na preferência para MAIL)
UID ou USERID	Identificador de usuários
CN	Nome Comum
T	Título
OU	Nome de Unidade Organizacional
DC	Componente de domínio
O	Nome da organização
STREET	Rua / Primeira linha do endereço
L	Nome da localidade
ST (ou SP ou S)	Nome do estado ou região

Nomes de atributos	Significado
PC	Código Postal / Código de Endereçamento Postal
C	País
UNSTRUCTUREDNAME	Nome do host
UNSTRUCTUREDADDRESS	endereço IP
DNQ	Qualificador de Nome Distinto

É possível usar o caractere curinga (*) no início ou no final do valor de atributos no lugar de qualquer número de caracteres. Por exemplo, para aceitar apenas certificados de qualquer pessoa com um nome que termine com `Smith` trabalhando para `IBM` em `GB`, digite:

```
CN=*Smith, O=IBM, C=GB
```

Por exemplo:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLPEER('CN=*Smith, O=IBM, C=GB')
```

Nota: Deve-se colocar a sequência SSLPEER entre aspas simples para assegurar que o caso de caractere seja mantido. Sem aspas simples, o IBM MQ transforma a sequência em maiúsculas.

Autenticando Partes que Iniciam Conexões com um Gerenciador de Filas

Quando outra parte inicia uma conexão ativada para TLS com um gerenciador de filas, este deve enviar seu certificado pessoal à parte iniciante como prova de identidade. Também é possível configurar opcionalmente o canal do gerenciador de filas, para que o gerenciador de filas recuse a conexão se a parte iniciante não enviar seu próprio certificado pessoal.

Para fazer isso, configure o atributo SSLCAUTH Este atributo é um atributo booleano e pode ter os valores OPTIONAL ou REQUIRED:

- OPTIONAL autentica o certificado de um cliente de conexão se um for fornecido, mas não requer que um cliente envie um. Um cliente será rejeitado se ele enviar um certificado inválido.
- REQUIRED rejeita quaisquer clientes de conexão que não forneçam um certificado TLS válido

Por exemplo:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCAUTH(REQUIRED)
```

IBM i Configurando as comunicações para SSL ou TLS no IBM i

As comunicações seguras que usam os protocolos de segurança criptográfica SSL ou TLS envolvem a configuração dos canais de comunicação e o gerenciamento de certificados digitais que serão usados para autenticação.

Para configurar sua instalação de SSL ou TLS, você deve definir seus canais para usar SSL ou TLS. Você deve também criar e gerenciar seus certificados digitais. Em alguns sistemas operacionais, é possível executar os testes com certificados autoassinados. No entanto, no IBM i, deve-se usar certificados pessoais assinados por uma CA local.

Para obter informações completas sobre a criação e o gerenciamento de certificados, consulte "Trabalhando com SSL/TLS no IBM i" na página 280.

Esta coleção de tópicos apresenta algumas das tarefas envolvidas na configuração de comunicações SSL ou TLS e fornece orientação passo a passo para concluir essas tarefas

Você talvez também deseje testar as autenticações de cliente SSL ou TLS, que são partes opcionais dos protocolos SSL e TLS. Durante o handshake de SSL ou TLS, o cliente SSL ou TLS sempre obtém e valida

um certificado digital do servidor. Com a implementação do IBM MQ , o servidor SSL ou TLS sempre solicita um certificado do cliente.

No IBM i, o cliente SSL ou TLS enviará um certificado somente se tiver um com o rótulo no formato correto do IBM MQ:

- Para um gerenciador de filas, `ibmwebspheremq`, seguido pelo nome de seu gerenciador de filas, mudado para minúsculas. Por exemplo, para `QM1`, `ibmwebspheremqqm1`.
- Para um IBM MQ C Client para IBM i, `ibmwebspheremq` seguido por seu logon ID do usuário alterado para minúscula, por exemplo `ibmwebspheremqmyuserid`.

O IBM MQ usa o prefixo `ibmwebspheremq` em um rótulo para evitar confusão com certificados de outros produtos. Assegure-se de especificar o rótulo inteiro do certificado em minúsculas.

O servidor SSL ou TLS sempre valida o certificado de cliente se um for enviado. Se o cliente SSL ou TLS não enviar um certificado, a autenticação falhará somente se a extremidade do canal que age como o servidor SSL ou TLS estiver definida com o parâmetro SSLCAUTH configurado como REQUIRED ou um valor de parâmetro SSLPEER configurado. Para obter mais informações, consulte Conectando dois gerenciadores de filas usando SSL ou TLS.

ALW Configurando as comunicações para SSL ou TLS no AIX, Linux, and Windows

As comunicações seguras que usam os protocolos de segurança criptográfica SSL ou TLS envolvem a configuração dos canais de comunicação e o gerenciamento de certificados digitais que serão usados para autenticação.

Para configurar sua instalação de SSL ou TLS, você deve definir seus canais para usar SSL ou TLS. Você deve também criar e gerenciar seus certificados digitais. Em sistemas AIX, Linux, and Windows, é possível executar os testes com certificados autoassinados.

Atenção: Não é possível usar uma combinação de certificados assinados por Curva elíptica e certificados assinados por RSA nos gerenciadores de filas que você deseja associar juntos usando canais ativados para TLS.

Os gerenciadores de filas que usam canais ativados para TLS devem usar todos os certificados assinados por RSA ou todos os certificados assinados pela EC, não uma mistura de ambos.

Consulte “Certificados digitais e compatibilidade de CipherSpec no IBM MQ” na página 48 para obter mais informações.

Certificados autoassinados não podem ser revogados, isso poderia permitir que um invasor copiasse uma identidade após o comprometimento de uma chave privada. CAs podem revogar um certificado comprometido, evitando seu uso adicional. O uso de certificados assinados por CA são, portanto, mais seguros em um ambiente de produção, embora certificados autoassinados sejam mais convenientes em um sistema de teste.

Para obter informações completas sobre a criação e o gerenciamento de certificados, consulte “Trabalhando com SSL/TLS no AIX, Linux, and Windows” na página 298.

Esta coleção de tópicos apresenta algumas das tarefas envolvidas na configuração de comunicações SSL, e fornece orientação passo a passo para concluir essas tarefas.

Você também pode querer testar a autenticação de cliente SSL ou TLS, que são uma parte opcional dos protocolos. Durante o handshake de SSL ou TLS, o cliente SSL ou TLS sempre obtém e valida um certificado digital do servidor. Com a implementação do IBM MQ , o servidor SSL ou TLS sempre solicita um certificado do cliente.

No AIX, Linux, and Windows, o cliente SSL ou TLS só enviará um certificado se ele tiver um rotulado no formato correto do IBM MQ:

- Para um gerenciador de filas, o formato é `ibmwebspheremq`, seguido pelo nome de seu gerenciador de filas mudado para minúsculas. Por exemplo, para `QM1`, `ibmwebspheremqqm1`

- Para um cliente IBM MQ, `ibmwebspheremq` seguido por seu ID do usuário de logon mudado para minúsculas, por exemplo, `ibmwebspheremqmyuserid`.

O IBM MQ usa o prefixo `ibmwebspheremq` em um rótulo para evitar confusão com certificados de outros produtos. Assegure-se de especificar o rótulo inteiro do certificado em minúsculas.

O servidor SSL ou TLS sempre valida o certificado de cliente se um for enviado. Se o cliente não enviar um certificado, a autenticação falhará somente se a extremidade do canal que age como o servidor SSL ou TLS estiver definida com o parâmetro SSLCAUTH configurado como REQUIRED ou um valor de parâmetro SSLPEER configurado. Para obter mais informações, consulte Conectando dois gerenciadores de filas usando SSL ou TLS.

z/OS Setting up communications for SSL or TLS on z/OS

Secure communications that use the SSL or TLS cryptographic security protocols involve setting up the communication channels and managing the digital certificates that you will use for authentication.

To set up your SSL or TLS installation you must define your channels to use SSL or TLS. You must also create and manage your digital certificates. On z/OS you can perform the tests with self-signed certificates, or with personal certificates signed by a local certificate authority (CA).

Self-signed certificates cannot be revoked, which could allow an attacker to spoof an identity after a private key has been compromised. CAs can revoke a compromised certificate, which prevents its further use. CA-signed certificates are therefore safer to use in a production environment, though self-signed certificates are more convenient for a test system.

For full information about creating and managing certificates, see “Working with SSL/TLS on z/OS” on page 312.

See the CERTLABL and CERTQSGL parameters of the ALTER QMGR command and the CERLABL parameter of the DEFINE CHANNEL command for more information.

The order of precedence is:

- Channel CERTLABL parameter
- QMGR CERTQSGL parameter if the channel is shared.

  For a sender channel, that means the transmission queue (XMITQ) is shared. For a receiver channel, that means the channel started through the shared listener, that is the listener with INDISP(GROUP).
- QMGR CERTLABL
- The default label of `ibmWebSphereMQ` followed by the name of the queue sharing group for shared channels, or the name of the queue manager.

This collection of topics introduces some of the tasks involved in setting up SSL or TLS communications, and provides step-by-step guidance on completing those tasks.

You might also want to test SSL or TLS client authentication, which are an optional part of the protocols. During the SSL or TLS handshake, the SSL or TLS client always obtains and validates a digital certificate from the server. With the IBM MQ implementation, the SSL or TLS server always requests a certificate from the client.

If the channel is shared, the channel first tries to find a certificate for the queue sharing group. If it does not find a certificate for a queue sharing group, it tries to find a certificate for the queue manager.

On z/OS, IBM MQ uses the `ibmWebSphereMQ` prefix on a label to avoid confusion with certificates for other products.

The SSL or TLS server always validates the client certificate if one is sent. If the SSL or TLS client does not send a certificate, authentication fails only if the end of the channel acting as the SSL or TLS server is defined with either the SSLCAUTH parameter set to REQUIRED or an SSLPEER parameter value set. For more information, see Connecting two queue managers using SSL or TLS.

Trabalhando com SSL/TLS

Estes tópicos fornecem instruções para executar tarefas únicas relacionadas ao uso do TLS com o IBM MQ.

Muitos deles são usados como etapas nas tarefas de alto nível descritos nas seguintes seções:

- “Identificando e autenticando usuários” na página 324
- “Autorizando o acesso aos objetos” na página 357
- “Confidencialidade das mensagens” na página 427
- “Integridade de dados de mensagens” na página 484
- “Mantendo Clusters Seguros” na página 485

IBM i Trabalhando com SSL/TLS no IBM i

Esta coleção de tópicos fornece instruções para tarefas individuais que trabalham com a Segurança da Camada de Transporte (TLS) no IBM MQ for IBM i.

Para o IBM i, o suporte ao TLS é integral para o sistema operacional. Assegure-se de que instalou os pré-requisitos listados na Requisitos de hardware e software no IBM i.

No IBM i, gerencie chaves e certificados digitais com a ferramenta DCM (Digital Certificate Manager).

Acessando DCM

Siga estas instruções para acessar a interface do DCM.

Sobre esta tarefa

Execute as etapas a seguir em um navegador da web que suporte quadros.

Procedimento

1. Acesse `http://`*`machine`*`.domain:2001` ou `https://`*`machine`*`.domain:2010`, em que *machine* é o nome do computador.
2. Digite um perfil do usuário e uma senha válidos quando solicitado.

   Assegure-se de que seu perfil do usuário tenha as autoridades especiais *ALLOBJ e *SECADM para permitir a criação de novos armazenamentos de certificados. Se você não possuir as autoridades especiais, poderá gerenciar apenas os certificados pessoais ou visualizar as assinaturas de objeto dos objetos que estão autorizados. Se você tiver autorização para usar um aplicativo de assinatura de objeto, também poderá assinar objetos a partir do DCM.
3. Na página Configurações da Internet, clique em **Digital Certificate Manager**.

   A página Digital Certificate Manager é exibida.

Atribuindo um certificado a um gerenciador de filas no IBM i

Use o DCM para designar um certificado a um gerenciador de filas.

Use o gerenciamento de certificado digital tradicional do IBM i para designar um certificado a um gerenciador de filas. Isso significa que é possível especificar que um gerenciador de filas use um armazenamento de certificados do sistema, e que o gerenciador de filas seja registrado para ser usado como um aplicativo com o Digital Certificate Manager. Para fazer isso, mude o valor do atributo **SSLKEYR** do gerenciador de filas para `*SYSTEM`.

Quando o parâmetro **SSLKEYR** é alterado para `*SYSTEM`, IBM MQ registra o gerenciador de fila como um aplicativo do servidor com um rótulo de aplicativo exclusivo de QIBM_WEBSPHERE_MQ_QMGRNAME e um rótulo com uma descrição de Qmgrname (WMQ). Observe que os atributos **CERTLABL** do canal não são usados se você usar o armazenamento de certificados *SYSTEM. O gerenciador de filas aparecerá como um aplicativo do servidor no Digital Certificate Manager e será possível atribuir a este aplicativo qualquer servidor ou certificado de cliente no armazenamento do sistema.

Como o gerenciador de filas é registrado como um aplicativo, recursos avançados do DCM tais como a definição das listas de confiança da CA poderão ser executadas.

Se o parâmetro **SSLKEYR** for alterado para um valor diferente de *SYSTEM, IBM MQ removerá o registro do gerenciador de filas como um aplicativo com Digital Certificate Manager. Se um gerenciador de filas for excluído, o registro também será removido do DCM. Um usuário com autoridade *SECADM suficiente também pode incluir ou remover manualmente os aplicativos do DCM.

Configurando um repositório de chaves no IBM i

Um repositório de chaves deve ser configurado em ambas as extremidades da conexão. Os armazenamentos de certificados padrão podem ser usados ou é possível criar seus próprios.

Uma conexão TLS requer um *repositório de chaves* em cada extremidade da conexão. Cada gerenciador de filas e IBM MQ MQI client devem ter acesso a um repositório de chaves. Se você deseja acessar o repositório de chaves usando um nome de arquivo e senha (ou seja, não usando a opção *SYSTEM) assegure que o perfil do usuário QMQM possua as seguintes autoridades:

- Autoridade de execução para o diretório que contém o repositório de chaves
- Autoridade de leitura para o arquivo que contém o repositório de chaves

Consulte a "O repositório de chaves SSL/TLS" na página 26 para obter mais informações. Observe que os atributos **CERTLABL** do canal não serão usados se você usar o armazenamento de certificados *SYSTEM..

No IBM i, os certificados digitais são armazenados em um armazenamento de certificados que é gerenciado com o DCM. Esses certificados digitais possuem rótulos que associam um certificado a um gerenciador de filas ou um IBM MQ MQI client. O TLS usa os certificados para propósitos de autenticação.

O rótulo é o valor do atributo **CERTLABL**, se ele estiver configurado ou o padrão `ibmwebspheremq` com o nome do gerenciador de filas ou o ID de logon do usuário do IBM MQ MQI client anexado, todo em minúsculas. Consulte Rótulos de certificado digital para obter detalhes.

O nome armazenamento de certificados do gerenciador de filas ou IBM MQ MQI client é composto de um caminho e nome de raiz. O caminho padrão é `/QIBM/UserData/ICSS/Cert/Server/` e o nome de raiz padrão é `Default`. No IBM i, o armazenamento de certificados padrão, `/QIBM/UserData/ICSS/Cert/Server/Default.kdb`, também é conhecido como `*SYSTEM` Como opção, é possível definir o seu próprio caminho e nome de raiz.

Se você definir seu próprio caminho ou nome de arquivo, configure as permissões para que o arquivo controle rigorosamente o acesso a ele.

"Alterando o local do repositório de chaves de um gerenciador de filas no IBM i" na página 284 traz informações sobre como especificar o nome do armazenamento de certificados. Você pode especificar o nome do armazenamento de certificados antes ou depois de criar o armazenamento de certificados.

Nota: As operações que você pode executar com o DCM podem ser limitadas pela autoridade do seu perfil de usuário. Por exemplo, as autoridades *ALLOBJ e *SECADM são necessárias para criar um certificado de CA.

IBM i *Criptografando senhas do repositório de chaves no IBM i*

Vários componentes do IBM MQ precisam de acesso a um repositório de chaves que contém certificados digitais ou chaves simétricas.. Um repositório de chaves é protegido por uma senha, pois ele contém informações confidenciais A senha do repositório de chaves deve ser armazenada em um local no qual o IBM MQ possa lê-lo quando o repositório de chaves for acessado A senha também deve ser criptografada para reduzir a probabilidade de acesso não autorizado para o repositório de chaves

Os seguintes componentes e recursos do IBM MQ suportam dois métodos diferentes para armazenar senhas do repositório de chaves:

- O repositório de chaves do TLS do gerenciador de fila
- IBM MQ MQI clients que usam o TLS

As senhas do repositório de chaves para uso por esses componentes são protegidas usando o sistema de proteção de senha IBM MQ . O mecanismo para fornecer uma senha e criptografá-la varia um pouco dependendo do componente:

O repositório de chaves TLS do gerenciador de filas

A senha é criptografada quando o atributo do gerenciador de filas **SSLKEYRPWD** é configurado usando o comando CHGMQM (Change Message Queue Manager) .

A senha é criptografada com o algoritmo AES-128 Os detalhes desse algoritmo são publicamente conhecidos e são considerados seguros.

A senha é armazenada em um arquivo stash em um formato proprietário que não é entendido por outro software que pode acessar o repositório de chaves.

Uma senha que é criptografada por um componente IBM MQ não pode ser usada por um componente IBM MQ diferente.

Uma chave de criptografia exclusiva pode ser fornecida quando a senha do repositório de chaves é criptografada Uma chave de criptografia exclusiva impede que qualquer pessoa que não tenha acesso à chave de criptografia possa decriptografar a senha. Você fornece essa chave por meio do atributo do gerenciador de filas do **INITKEY** , que deve ser configurado antes de fornecer uma senha a ser criptografada

Para obter mais informações sobre o sistema de proteção de senha do IBM MQ , consulte “Protegendo senhas em arquivos de configuração do componente do IBM MQ” na página 574

IBM MQ MQI clients que usam TLS

O “Utilitário do cliente SSL (amqrsslc) do IBM MQ para IBM i” na página 296 pode armazenar a senha do repositório de chaves em um arquivo stash Consulte também Administrando usando comandos MQSC no IBM i.

A senha é criptografada com o algoritmo AES-128 Os detalhes desse algoritmo são publicamente conhecidos e são considerados seguros.

A senha é armazenada em um arquivo stash em um formato proprietário que não é entendido por outro software que pode acessar o repositório de chaves.

Uma chave de criptografia exclusiva pode ser fornecida quando a senha do repositório de chaves é criptografada Uma chave de criptografia exclusiva impede que qualquer pessoa que não tenha acesso à chave de criptografia possa decriptografar a senha. Você fornece essa chave por meio do parâmetro **-sf**

A senha criptografada é armazenada em um arquivo stash no mesmo diretório que o arquivo do repositório de chaves

O IBM MQ MQI clients também suporta senhas fornecidas por outros mecanismos. Consulte “Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on IBM i” na página 286.

Independentemente do método escolhido para criptografar a senha do repositório de chaves, assegure-se de estar ciente das limitações da criptografia de senhas armazenadas. Consulte “Os limites para proteção por meio de criptografia de senha” na página 581.

Conceitos relacionados

“Fornecendo a senha do repositório de chaves para um gerenciador de filas no IBM i” na página 285
Como o repositório de chaves contém informações sensíveis, ele é protegido com uma senha. Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

“Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on IBM i” na página 286
Como o repositório de chaves contém informações sensíveis, ele é protegido com uma senha. Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

“Trabalhando com SSL/TLS no IBM i” na página 280

Esta coleção de tópicos fornece instruções para tarefas individuais que trabalham com a Segurança da Camada de Transporte (TLS) no IBM MQ for IBM i.

Criando um armazenamento de certificados no IBM i
Se não desejar usar o armazenamento de certificados padrão, siga este procedimento para criar seu próprio.

Sobre esta tarefa

Crie um novo armazenamento de certificados somente se você não desejar usar o certificado padrão de armazenamento do IBM i.

Para especificar que o armazenamento de certificados do sistema IBM i deve ser usado, altere o valor do atributo SSLKEYR do gerenciador de filas para *SYSTEM. Esse valor indica que o gerenciador de filas usa o armazenamento de certificados do sistema e que o gerenciador de filas está registrado para ser usado como um aplicativo com o Digital Certificate Manager (DCM).

Procedimento

1. Acesse a interface DCM, conforme descrito em “Acessando DCM” na página 280
2. No painel de navegação, clique em **Criar um Novo Armazenamento de Certificados**.
   A página Criar Novo Armazenamento de Certificados é exibida no quadro de tarefas.
3. No quadro de tarefas, selecione **Outro Armazenamento de Certificados do Sistema** e clique em **Continuar**.
   A página Criar um Certificado no Novo Armazenamento de Certificados é exibida no quadro de tarefas.
4. Selecione **Não - Não criar um certificado no armazenamento de certificados** e clique em **Continuar**.
   A página Nome e Senha do Armazenamento de Certificados é exibida no quadro de tarefas.
5. No campo **Caminho e nome de arquivo do armazenamento de certificados**, digite um caminho e nome de arquivo IFS, por exemplo `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`
6. Digite uma senha no campo **Senha** e digite-a novamente no campo **Confirmar Senha**. Clique em **Continue**.
   Anote a senha (que faz distinção entre maiúsculas e minúsculas) porque ela será necessária quando você armazenar a chave do repositório em arquivo stash.
7. Para sair do DCM, feche a janela do navegador.

Como proceder a seguir

Quando você tiver criado o armazenamento de certificados usando o DCM, assegure-se de armazenar a senha em arquivo stash, conforme descrito em “Armazenar em arquivo stash a senha do armazenamento de certificados em sistemas IBM i” na página 283.

Tarefas relacionadas

“Importando um certificado em um repositório de chaves no IBM i” na página 294
Siga este procedimento para importar um certificado.

Armazenar em arquivo stash a senha do armazenamento de certificados em sistemas IBM i
Armazenar em arquivo stash a senha de armazenamento de certificados usando comandos CL.

As instruções a seguir se aplicam a armazenar em arquivo stash a senha de armazenamento de certificados no IBM i para um gerenciador de filas. Como alternativa, para um IBM MQ MQI client, se você não estiver usando o armazenamento de certificados *SYSTEM (ou seja, o ambiente MQSSLKEYR estiver configurado para um valor diferente de *SYSTEM) siga o procedimento descrito na seção “Armazenar em arquivo stash a senha de armazenamento de certificados” na página 297 de “Utilitário do cliente SSL (amqrsslc) do IBM MQ para IBM i” na página 296..

Se você tiver especificado que o armazenamento de certificados *SYSTEM deve ser usado (ao alterar o valor do atributo SSLKEYR do gerenciador de filas para *SYSTEM) não se deve seguir estas etapas.

Ao criar o armazenamento de certificados usando o DCM, use os seguintes comandos para proteger a senha:

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

A senha faz distinção entre maiúsculas e minúsculas. Ela deve ser inserida entre aspas simples exatamente como você a inseriu na etapa 6 em "Criando um armazenamento de certificados no IBM i" na página 283.

Nota: Se você não estiver usando o armazenamento de certificados do sistema padrão e não armazenar a senha em arquivo stash, as tentativas para iniciar os canais TLS falharão porque eles não podem obter a senha necessária para acessar o armazenamento de certificados.

Proteção de Senha

Quando uma senha do repositório de chave é especificada, o IBM MQ criptografa a senha usando o sistema IBM MQ Password Protection. Para criptografar a senha, uma chave inicial é usada; se ela não for fornecida para o gerenciador de fila, uma chave padrão será usada.

Antes de fornecer a senha do repositório de chaves, você deve configurar uma chave inicial exclusiva para o gerenciador de fila.. É possível fazer isso usando o atributo **INITKEY** do comando MQSC **ALTER QMGR** :

```
ALTER QMGR INITKEY('value')
```

Localizando o repositório de chaves para um gerenciador de filas no IBM i

Use este procedimento para obter o local do armazenamento de certificados do gerenciador de filas.

Procedimento

1. Exiba os atributos de seu gerenciador de filas, usando o seguinte comando:

   ```
   DSPMQM MQMNAME('queue manager name')
   ```

2. Examine a saída do comando para o nome do caminho e de raiz do armazenamento de certificados.

   Por exemplo: `/QIBM/UserData/ICSS/Cert/Server/Default`, em que `/QIBM/UserData/ICSS/Cert/Server` é o caminho e `Default` é o nome derivado.

Alterando o local do repositório de chaves de um gerenciador de filas no IBM i

Altere o local do armazenamento de certificados do gerenciador de filas usando CHGMQM ou ALTER QMGR.

Procedimento

Use o comando CHGMQM ou o comando ALTER QMGR MQSC para configurar o atributo de repositório de chaves do gerenciador de filas.

a) Usando CHGMQM: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

b) Usando ALTER QMGR: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

Em qualquer um dos casos, o armazenamento de certificados tem o nome completo do arquivo: `/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb`

Como proceder a seguir

Ao alterar o local de um armazenamento de certificados do gerenciador de filas, os certificados não serão transferidos a partir do local antigo. Se os certificados de CA pré-instalados ao criar o armazenamento de certificados forem insuficientes, você deverá preencher o novo armazenamento de certificados com

certificados, conforme descrito em “Importando um certificado em um repositório de chaves no IBM i” na página 294. Também é necessário proteger a senha para o novo local, conforme descrito em “Armazenar em arquivo stash a senha do armazenamento de certificados em sistemas IBM i” na página 283.

IBM i *Fornecendo a senha do repositório de chaves para um gerenciador de filas no IBM i*

Como o repositório de chaves contém informações sensíveis, ele é protegido com uma senha. Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

IBM MQ fornece um mecanismo para fornecer a senha do repositório de chaves para um gerenciador de filas:

- O parâmetro **SSLKEYRPWD** no comando **CHGMQM**

A senha do repositório de chaves é criptografada usando o sistema de proteção de senha do IBM MQ Para obter mais informações sobre os métodos de proteção da senha do repositório de chaves, consulte “Criptografando senhas do repositório de chaves no IBM i” na página 281

Consulte também Administrando usando comandos MQSC no IBM i.

O atributo SSLKEYRPWD

Para fornecer uma senha do repositório de chaves diretamente para o gerenciador de filas, execute o comando **CHGMQM** a seguir, substituindo *queue_manager* pelo nome do gerenciador de filas e *password* pela senha do repositório de chaves:

```
CHGMQM MQMNAME('queue_manager') SSLKEYRPWD('password')
```

Atenção: Assegure-se de colocar o nome do gerenciador de filas e a senha entre aspas simples, caso contrário, IBM MQ converterá os caracteres em maiúsculas.

Quando uma senha do repositório de chaves é especificada usando esse método, a senha é criptografada usando o sistema de proteção de senha do IBM MQ antes de ser armazenada

Uma chave de criptografia, conhecida como a chave inicial, é usada para criptografar a senha. Configure o gerenciador para usar uma chave inicial exclusiva para proteger com segurança a senha. Se você não fornecer uma chave inicial, a chave padrão será usada

Assegure que o gerenciador de filas esteja configurado com uma chave inicial exclusiva antes de configurar a senha do repositório de chaves. É possível modificar a chave inicial usando o atributo **INITKEY** no comando **ALTER QMGR** .. Por exemplo:

```
ALTER QMGR INITKEY('mykey')
```

Aviso: Se você modificar a chave inicial depois de configurar a senha do repositório de chaves, a senha do repositório de chaves não será criptografada com a nova chave inicial Se você alterar a chave inicial, também deverá reconfigurar a senha do repositório de chaves. Caso contrário, o IBM MQ não poderá decriptografar a senha do repositório de chaves e, portanto, não poderá acessar o repositório de chaves

Para obter mais informações sobre o atributo **SSLKEYRPWD** , consulte O parâmetro **SSLKEYRPWD** no comando **CHGMQM**

Conceitos relacionados

“Criptografando senhas do repositório de chaves no IBM i” na página 281
Vários componentes do IBM MQ precisam de acesso a um repositório de chaves que contém certificados digitais ou chaves simétricas.. Um repositório de chaves é protegido por uma senha, pois ele contém informações confidenciais A senha do repositório de chaves deve ser armazenada em um local no qual o IBM MQ possa lê-lo quando o repositório de chaves for acessado A senha também deve ser criptografada para reduzir a probabilidade de acesso não autorizado para o repositório de chaves

“Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on IBM i” na página 286

Como o repositório de chaves contém informações sensíveis, ele é protegido com uma senha. Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

IBM i *Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on IBM i*

Como o repositório de chaves contém informações sensíveis, ele é protegido com uma senha. Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

O IBM MQ fornece quatro mecanismos para fornecer a senha do repositório de chaves para um IBM MQ MQI client:

- “Os campos KeyRepoPassword do MQSCO ” na página 286
- “A variável de ambiente MQKEYRPWD” na página 286
- “O atributo SSLKeyRepositoryPassword do arquivo de configuração do cliente” na página 287
- “O arquivo stash do repositório de chave” na página 287

Se você não usar um arquivo stash de repositório de chave, será possível fornecer a senha do repositório de chave como uma sequência de texto simples ou uma sequência que é criptografada usando o sistema de proteção de senha IBM MQ Para obter mais informações sobre os métodos de proteção da senha do repositório de chaves, consulte “Criptografando senhas do repositório de chaves no IBM i” na página 281

Os campos KeyRepoPassword do MQSCO

Para fornecer uma senha do repositório de chaves usando a estrutura MQSCO, deve-se usar uma combinação dos três campos de sequência de variáveis a seguir:.

KeyRepoPasswordLength
O comprimento da senha.

KeyRepoPasswordPtr
Um ponteiro para o local na memória que contém a senha

KeyRepoPasswordOffset
O local da senha na memória, representado como o número de bytes do início da estrutura MQSCO.

Nota: É possível fornecer apenas um de **KeyRepoPasswordPtr** ou **KeyRepoPasswordOffset**

Por exemplo:

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

Atenção: Se você fornecer a senha usando esse método, criptografe a senha antes que ela seja fornecida para o aplicativo IBM MQ client Para obter mais informações, consulte “Criptografando a senha do repositório de chaves” na página 287.

Para obter mais informações sobre a estrutura MQCSO, consulte Opções de configuração de MQSCO-SSL/TLS

A variável de ambiente *MQKEYRPWD*

Se uma senha do repositório de chave não for fornecida ao cliente usando a estrutura MQSCO, será possível especificar a senha do repositório de chaves usando a variável de ambiente *MQKEYRPWD* . Por exemplo:

```
export MQKEYRPWD=passw0rd
```

ou

```
set MQKEYRPWD=passw0rd
```

em que *passw0rd* é a sua senha

Atenção: Se você fornecer a senha usando esse método, criptografe a senha antes de configurar o valor da variável de ambiente.. Para obter informações adicionais, consulte “Criptografando a senha do repositório de chaves” na página 287.

O atributo SSLKeyRepositoryPassword do arquivo de configuração do cliente

Se uma senha do repositório de chaves não for fornecida para o cliente usando um dos outros métodos, será possível especificar a senha do repositório de chaves usando o atributo **SSLKeyRepositoryPassword** na sub-rotina **SSL** do arquivo de configuração do cliente Por exemplo:

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

Atenção: Se você fornecer a senha usando esse método, criptografe a senha antes de configurar o valor do atributo **SSLKeyRepositoryPassword** .. Para obter mais informações, consulte “Criptografando a senha do repositório de chaves” na página 287.

Para obter mais informações sobre a sub-rotina SSL do arquivo de configuração do cliente, consulte Sub-rotina SSL do arquivo de configuração do cliente

O arquivo stash do repositório de chave

Se a senha do repositório de chave não for fornecida ao cliente usando um dos outros métodos, o IBM MQ assumirá que um arquivo stash existe no mesmo diretório que o repositório de chave. O arquivo stash tem o mesmo nome de raiz que o repositório de chaves, mas tem a extensão `.sth`

Um arquivo stash de repositório de chaves é criado usando a ferramenta de linha de comando **amqrsslc** Para criar o arquivo stash, execute o comando a seguir:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

Este comando solicita a senha para criptografar. A senha é criptografada pelo sistema de proteção de senha do IBM MQ , com uma chave de criptografia padrão, a menos que uma seja fornecida usando o parâmetro **-sf**

Para obter mais informações, veja “Utilitário do cliente SSL (amqrsslc) do IBM MQ para IBM i” na página 296 e “Criptografando a senha do repositório de chaves” na página 287.

Criptografando a senha do repositório de chaves

Se você fornecer a senha do repositório de chaves usando qualquer método diferente de um arquivo stash, criptografe a senha usando o sistema de proteção de senha IBM MQ . Para criptografar a senha, execute o comando **runmqicred** .. Insira a senha do repositório de chaves quando solicitado.. O comando gera a senha criptografada.. A senha criptografada pode ser fornecida para o IBM MQ MQI client em vez da senha de texto simples usando qualquer um dos métodos descritos.

Uma chave de criptografia, conhecida como a chave inicial, é usada para criptografar a senha. Quando você criptografar a senha, use uma chave inicial exclusiva para proteger com segurança a senha Para fornecer sua própria chave inicial, use o parâmetro **-sf** no comando **runmqicred** . Se você não fornecer uma chave inicial, a chave padrão será usada

Para obter mais informações, consulte runmqicred (proteger as senhas do cliente IBM MQ).

Se você fornecer a sua própria chave inicial quando a senha do repositório de chaves for criptografada e fornecer a senha criptografada para o IBM MQ MQI client, também deverá assegurar que você forneça a mesma chave inicial para o IBM MQ MQI client Para obter mais informações sobre como fornecer a chave

inicial para um IBM MQ MQI client, consulte “Fornecendo uma chave inicial para um IBM MQ MQI client em IBM i” na página 288

Conceitos relacionados

“Criptografando senhas do repositório de chaves no IBM i” na página 281
Vários componentes do IBM MQ precisam de acesso a um repositório de chaves que contém certificados digitais ou chaves simétricas.. Um repositório de chaves é protegido por uma senha, pois ele contém informações confidenciais A senha do repositório de chaves deve ser armazenada em um local no qual o IBM MQ possa lê-lo quando o repositório de chaves for acessado A senha também deve ser criptografada para reduzir a probabilidade de acesso não autorizado para o repositório de chaves

“Fornecendo a senha do repositório de chaves para um gerenciador de filas no IBM i” na página 285
Como o repositório de chaves contém informações sensíveis, ele é protegido com uma senha. Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

IBM i *Fornecendo uma chave inicial para um IBM MQ MQI client em IBM i*

Se você fornecer variáveis para um IBM MQ MQI client que tenha sido criptografado usando o IBM MQ Password Protection System, poderá ser necessário fornecer a chave inicial correspondente que foi usada para criptografar o valor

Se você não especificou uma chave inicial ao criptografar o valor, não será necessário fornecer nenhum valor de chave inicial para o IBM MQ client.. No entanto, se você usou uma chave inicial exclusiva, será possível fornecer a chave inicial para o IBM MQ client usando os métodos a seguir:

- “Fornecendo a chave inicial usando a estrutura MQCSP” na página 288
- “Fornecendo a chave inicial usando a variável de ambiente MQS_MQI_KEYFILE” na página 288
- “Fornecendo a chave inicial usando o arquivo de configuração do cliente” na página 289

Fornecendo a chave inicial usando a estrutura MQCSP

Para fornecer a chave inicial usando a estrutura MQCSP, você deve usar uma combinação dos três campos de sequência de variáveis a seguir:

InitialKeyLength
O comprimento da chave inicial

InitialKeyPtr
Um ponteiro para o local na memória contendo a chave inicial

InitialKeyOffset
O local da chave inicial na memória, representado como o número de bytes do início da estrutura MQCSP.

Nota: É possível fornecer apenas um de **InitialKeyPtr** ou **InitialKeyOffset**

Por exemplo:

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

Fornecendo a chave inicial usando a variável de ambiente MQS_MQI_KEYFILE

Se uma chave inicial não for fornecida ao cliente usando a estrutura MQCSP, IBM MQ verificará a variável de ambiente *MQS_MQI_KEYFILE* . Você deve configurar essa variável de ambiente para o local de um arquivo contendo uma única linha de texto, consistindo na chave inicial que deseja usar.

Por exemplo, se um arquivo chamado `mykey.key` existir no diretório raiz e contiver a chave inicial, você deverá configurar a variável de ambiente da seguinte forma:

```
export MQS_MQI_KEYFILE=/mykey.key
```

ou

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

Fornecendo a chave inicial usando o arquivo de configuração do cliente

Se uma chave inicial não for fornecida ao cliente usando um mecanismo anterior, o IBM MQ verificará o atributo **MQIInitialKeyFile** da sub-rotina Segurança do arquivo `mqclient.ini`. Você deve configurar esse atributo para o local de um arquivo contendo uma única linha de texto, consistindo na chave inicial que deseja usar.

Por exemplo, se um arquivo chamado `mykey.key` existir no diretório raiz e contiver a chave inicial, o arquivo de configuração do cliente deverá conter o seguinte:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

Conceitos relacionados

“Criptografando senhas do repositório de chaves no IBM i” na página 281
Vários componentes do IBM MQ precisam de acesso a um repositório de chaves que contém certificados digitais ou chaves simétricas.. Um repositório de chaves é protegido por uma senha, pois ele contém informações confidenciais A senha do repositório de chaves deve ser armazenada em um local no qual o IBM MQ possa lê-lo quando o repositório de chaves for acessado A senha também deve ser criptografada para reduzir a probabilidade de acesso não autorizado para o repositório de chaves

“Trabalhando com SSL/TLS no IBM i” na página 280
Esta coleção de tópicos fornece instruções para tarefas individuais que trabalham com a Segurança da Camada de Transporte (TLS) no IBM MQ for IBM i.

Criando uma autoridade de certificação e certificado para testes no IBM i

Use este procedimento para criar um certificado de CA local para assinar solicitações de certificados e para criar e instalar o certificado de CA.

Antes de começar

As instruções deste tópico supõem que não existe uma autoridade de certificação (CA) local. Se existir uma CA local, acesse “Solicitando um certificado do servidor no IBM i” na página 290.

Sobre esta tarefa

Os certificados de CA fornecidos ao instalar o TLS são assinados pela CA emitente. No IBM i, é possível gerar uma autoridade de certificação local que pode assinar certificados do servidor para testar comunicações TLS em seu sistema. Siga estas etapas em um navegador da web para criar um certificado de CA local:

Procedimento

1. Acesse a interface DCM, conforme descrito em “Acessando DCM” na página 280.
2. No painel de navegação, clique em **Criar uma Autoridade de Certificados**.

   A página Criar uma Autoridade de Certificação é exibida no quadro de tarefas.
3. Digite uma senha no campo **Senha de Armazenamento de Certificados** e digite-a novamente no campo **Confirmar Senha**.
4. Digite um nome no campo **Nome da autoridade de certificação (CA)** , por exemplo `TLS Test Certificate Authority`.

5. Digite valores apropriados nos campos **Nome Comum** e **Organização** e selecione um país. Para os campos opcionais restantes, digite os valores que você precisa.
6. Digite um período de validade para o CA local no campo **Período de Validade**.

   O valor padrão é 1095 dias.
7. Clique em **Continue**.

   O CA é criado e o DCM criará um armazenamento de certificados e um certificado de CA para o CA local.
8. Clique em **Instalar Certificado**.

   A caixa de diálogo Gerenciador de Download é exibida.
9. Digite o nome do caminho completo para o arquivo temporário no qual deseja armazenar o certificado de CA e clique em **Salvar**.
10. Quando o download estiver concluído, clique em **Abrir**.

    A janela Certificado é exibida.
11. Clique em **Instalar Certificado**.

    O Assistente de Importação de Certificado é exibido.
12. Clique em **Avançar**.
13. Selecione **Selecionar o armazenamento de certificados automaticamente com base no tipo de certificado** e clique em **Avançar**.
14. Clique em **Finish**.

    Uma janela de confirmação é exibida.
15. Clique em **OK**.
16. Na janela Certificado, clique em **OK**.
17. Clique em **Continue**.

    A página Política de Autoridade de Certificação é exibida no quadro de tarefas.
18. No campo **Permitir a criação de certificados de usuários**, selecione **Sim**.
19. No campo **Período de Validade**, digite o período de validade dos certificados que são emitidos pela sua CA local.

    O valor padrão é 365 dias.
20. Clique em **Continue**.

    A página Criar um Certificado no Novo Armazenamento de Certificados é exibida no quadro de tarefas.
21. Verifique se nenhum dos aplicativos está selecionado.
22. Clique em **Continuar** para concluir a configuração da CA local.

Como proceder a seguir

Se você precisar renovar um certificado existente, consulte Renomeando um certificado existente na documentação do IBM i

Solicitando um certificado do servidor no IBM i

Os certificados digitais são protegidos contra personificação, certificando que uma chave pública pertence a uma entidade especificada. Um novo certificado do servidor pode ser solicitado a partir de uma autoridade de certificação usando o Digital Certificate Manager (DCM).

Sobre esta tarefa

Execute as seguintes etapas em um navegador da web:

Procedimento

1. Acesse a interface DCM, conforme descrito em “Acessando DCM” na página 280.

2. No painel de navegação, clique em **Selecionar um Armazenamento de Certificados**.

   A página Selecionar um Armazenamento de Certificados é exibida no quadro de tarefas.
3. Selecione o armazenamento de certificados que deseja usar e clique em **Continuar**.
4. Opcional: Se você tiver selecionado ***SYSTEM** na etapa 3, insira a senha de armazenamento do sistema e clique em **Continuar**.
5. Opcional: Se você tiver selecionado **Outro armazenamento de certificados do sistema** na etapa 3, no campo **Caminho e nome do arquivo do armazenamento de certificados**, digite o caminho e nome de arquivo IFS configurado durante a criação do armazenamento de certificados. Além disso, digite uma senha no campo **Senha do armazenamento de certificados**. Em seguida, clique em **Continuar**
6. No painel de navegação, clique em **Criar Certificado**.
7. No quadro de tarefas, selecione o botão de opção **Certificado do servidor ou do cliente** e clique em **Continuar**.

   A página Selecionar uma Autoridade de Certificação (CA) é exibida no quadro de tarefas.
8. Se você possuir uma CA local na sua estação de trabalho, escolha a CA local ou uma CA comercial para assinar o certificado. Selecione o botão de opção para a CA desejada e clique em **Continuar**.

   A página Criar um Certificado é exibida no quadro de tarefas.
9. Opcional: Para um gerenciador de filas, no campo **Rótulo do certificado** insira o rótulo certificado.

   O rótulo é o valor do atributo **CERTLABL**, se ele estiver configurado ou o padrão `ibmwebspheremq` com o nome do gerenciador de filas anexado, tudo em minúsculas. Consulte Rótulos de certificado digital para obter detalhes.

   Por exemplo, para o gerenciador de filas QM1, digite `ibmwebspheremqqm1` para usar o valor padrão.
10. Opcional: Para um IBM MQ MQI client, no campo **Certificado do rótulo**, digite `ibmwebspheremq` seguido por seu ID do usuário de logon em letras minúsculas.
    Por exemplo, digite `ibmwebspheremqmyuserID`
11. Digite valores apropriados nos campos **Nome Comum** e **Organização** e selecione um país. Para os campos opcionais restantes, digite os valores que você precisa.

Resultados

Se você selecionou uma CA comercial para assinar o certificado, o DCM criará um pedido de certificado no formato PEM (Privacy-Enhanced Mail). Encaminhe o pedido para a CA escolhida.

Se você selecionou a CA local para assinar o certificado, o DCM informará que o certificado foi criado no armazenamento de certificados e poderá ser usado.

Solicitando um certificado de servidor para um sistema remoto emIBM i

Siga este procedimento para criar um certificado assinado pela sua autoridade de certificação (CA) local ou para solicitar um certificado de servidor assinado por uma CA comercial para importação para um repositório de chaves em outras plataformas.

Sobre esta tarefa

Um certificado de usuário deve ser usado quando o Digital Certificate Manager (DCM) servir como o gerenciador de certificado para o IBM MQ em várias plataformas. Para certificados pessoais distribuídos para outras plataformas e importados para um repositório de chaves, execute as seguintes etapas em um navegador da web:

Procedimento

1. Acesse a interface DCM, conforme descrito em “Acessando DCM” na página 280.
2. Na área de janela de **navegação**, clique em **Criar Certificado**.

   A página **Criar Certificado** é exibida no quadro de tarefas.
3. No painel **Criar Certificado**, selecione o botão de opções **Certificado de Usuário** e clique em **Continuar**.

A página **Criar Certificado de Usuário** é exibida.

4. No painel **Criar Certificado de Usuário**, preencha os campos obrigatórios de Informações de Certificado para **Nome da organização**, **Estado** ou **município**, **País** ou **região**. Opcionalmente, coloque valores nos campos **Unidade de Organização** e **Localidade** ou **cidade**. Clique em **Continue**.

   O **Nome comum** é configurado automaticamente para o ID do usuário com o qual você efetuou logon no sistema iSeries.
5. No próximo painel **Criar Certificado de Usuário**, clique em **Instalar Certificado** e clique em **Continuar**.

   É exibida uma mensagem indicando: `Seu certificado pessoal foi instalado. É necessário manter uma cópia de backup desse certificado.`
6. Clique em **OK**.
7. Dependendo do navegador da Web usado para acessar o DCM, execute uma das etapas a seguir:
   - Para o Microsoft Edge escolha: **Ferramentas>Opções da Internet>Guia Conteúdo>botão Certificados>Guia Pessoal>**. Selecione o certificado e clique em **Exportar**.
   - Para Mozilla Firefox escolha: **Ferramentas>Opções>Avançado>guia Criptografia>botão Visualizar certificados>guia Seus certificados>**. Selecione o certificado e clique em **Backup**. Selecione o caminho e o nome do arquivo e clique em **OK**.
8. Transfira o certificado exportado para o sistema remoto usando FTP no formato binário.
9. Importe o certificado que foi exportado na etapa"7" na página 292 para o repositório de chaves no sistema remoto.
   - Se o certificado foi salvo usandoMicrosoft Edge, use as instruções descritas em"Importando um certificado pessoal a partir de um arquivo .pfx Microsoft" na página 564 arquivo.
   - Se o certificado tiver sido salvo usando o Mozilla Firefox, use as instruções descritas em Importando um Certificado Pessoal para um Repositório de Chaves.

   Durante a importação, certifique-se de que o nome do rótulo do certificado pessoal e do certificado de assinante sejam alterados para o valor queIBM MQ espera. O rótulo deve ser o valor doIBM MQ gerenciador de filas**CERTLABL** atributo, se estiver definido, ou o valor padrão de`ibmwebspheremq` com o nome do gerenciador de filas anexado, tudo em letras minúsculas. Para mais informações, veja Etiquetas de certificados digitais .

Incluindo certificados de servidor em um repositório de chaves no IBM i

Siga este procedimento para incluir um certificado solicitado no repositório de chaves.

Sobre esta tarefa

Depois que a CA enviar um novo certificado do servidor, inclua-o ao armazenamento de certificados a partir do qual o pedido foi gerado. Se a CA enviar o certificado como parte de uma mensagem de e-mail, copie o certificado em um arquivo separado.

Nota:

- Não será necessário executar esse procedimento se o certificado do servidor for assinado pela sua CA local.
- Antes de importar um certificado do servidor no formato PKCS #12 para o DCM, é necessário primeiramente importar o certificado de CA correspondente.

Use o seguinte procedimento para receber um certificado do servidor para o armazenamento de certificados do gerenciador de filas:

Procedimento

1. Acesse a interface DCM, conforme descrito em "Acessando DCM" na página 280.
2. Na categoria de tarefa **Gerenciar Certificados** no painel de navegação, clique em **Importar Certificado**.

A página Importar Certificado é exibida no quadro de tarefas.

3. Selecione o botão de opção para o seu tipo de certificado e clique em **Continuar**.

   A página Servidor de Importação ou Certificado do Cliente, ou a página Importar Certificado CA (Autoridade de Certificação) é exibida no quadro de tarefas.

4. No campo **Importar Arquivo**, digite o nome do arquivo do certificado que deseja importar e clique em **Continuar**.

   O DCM determina automaticamente o formato do arquivo.

5. Se o certificado for um certificado **Servidor ou cliente**, digite a senha no quadro de tarefas e clique em **Continuar**.

   O DCM informará que o certificado foi importado.

Exportando um certificado de um repositório de chaves no IBM i

Exportar um certificado exporta as chaves pública e privada. Esta ação deve ser tomada com extremo cuidado, pois passar uma chave privada comprometeria completamente a sua segurança.

Antes de começar

Quando você compartilha um certificado do usuário com outro usuário, você troca chaves públicas. Este processo é descrito na **Tarefa 5. Certificados de compartilhamento do** na seção Certificados de compartilhamento de "Guia de iniciação rápida para o AMS no AIX and Linux" na página 624. Quando você exporta um certificado conforme descrito aqui, você exporta as chaves pública e privada. Esta ação deve ser tomada com extremo cuidado, pois passar uma chave privada comprometeria completamente a sua segurança.

Sobre esta tarefa

Execute as seguintes etapas no computador do qual você deseja exportar o certificado:

Procedimento

1. Acesse a interface DCM, conforme descrito em "Acessando DCM" na página 280.
2. No painel de navegação, clique em **Selecionar um Armazenamento de Certificados**.

   A página Selecionar um Armazenamento de Certificados é exibida no quadro de tarefas.

3. Selecione o armazenamento de certificados que deseja usar e clique em **Continuar**.
4. Opcional: Se você tiver selecionado ***SYSTEM** na etapa 3, insira a senha de armazenamento do sistema e clique em **Continuar**.
5. Opcional: Se você tiver selecionado **Outro Armazenamento de Certificados do Sistema** na etapa 3, no campo **Caminho e Nome de Arquivo do Armazenamento de Certificados**, digite o caminho e o nome de arquivo do IFS configurados durante a criação do armazenamento de certificados e digite uma senha no campo **Senha de Armazenamento do Certificado**. Em seguida, clique em **Continuar**
6. Na categoria de tarefa **Gerenciar Certificados**, no painel de navegação, clique em **Exportar Certificado**.

   A página Exportar um Certificado é exibida no quadro de tarefas.

7. Selecione o botão de opção para o seu tipo de certificado e clique em **Continuar**.

   A página Servidor de Exportação ou Certificado do Cliente ou a página Exportar Certificado da Autoridade de Certificação (CA) é exibida no quadro de tarefas.

8. Selecione o certificado que deseja exportar.
9. Selecione o botão de opção para especificar se deseja exportar o certificado para um arquivo ou diretamente para outro armazenamento de certificados.
10. Se você selecionou exportar um certificado de servidor ou cliente para um arquivo, forneça as seguintes informações:

    - O caminho e o nome do arquivo do local onde deseja armazenar o certificado exportado.

- Para um certificado pessoal, a senha que é usada para criptografar o certificado exportado e o release de destino. Para certificados de CA, não é necessário especificar a senha.

11. Se você selecionou exportar um certificado diretamente para outro armazenamento de certificados, especifique o armazenamento de certificados de destino e a senha.
12. Clique em **Continue**.

Importando um certificado em um repositório de chaves no IBM i

Siga este procedimento para importar um certificado.

Antes de começar

Antes de importar um certificado pessoal no formato PKCS #12 para o DCM, é necessário primeiramente importar o certificado de CA correspondente.

Sobre esta tarefa

Execute estas etapas na máquina para a qual deseja importar o certificado.

Procedimento

1. Acesse a interface DCM, conforme descrito em "Acessando DCM" na página 280.
2. No painel de navegação, clique em **Selecionar um Armazenamento de Certificados**.

   A página Selecionar um Armazenamento de Certificados é exibida no quadro de tarefas.
3. Selecione o armazenamento de certificados que deseja usar e clique em **Continuar**.
4. Opcional: Se você tiver selecionado ***SYSTEM** na etapa 3, insira a senha de armazenamento do sistema e clique em **Continuar**.
5. Opcional: Se você tiver selecionado **Outro Armazenamento de Certificados do Sistema** na etapa 3, no campo **Caminho e Nome de Arquivo do Armazenamento de Certificados**, digite o caminho e o nome de arquivo do IFS configurados durante a criação do armazenamento de certificados e digite uma senha no campo **Senha de Armazenamento do Certificado**. Em seguida, clique em **Continuar**
6. Na categoria de tarefa **Gerenciar Certificados** no painel de navegação, clique em **Importar Certificado**.

   A página Importar Certificado é exibida no quadro de tarefas.
7. Selecione o botão de opção para o seu tipo de certificado e clique em **Continuar**.

   A página Servidor de Importação ou Certificado do Cliente ou a página Importar Certificado da Autoridade de Certificação (CA) é exibida no quadro de tarefas.
8. No campo **Importar Arquivo**, digite o nome do arquivo do certificado que deseja importar e clique em **Continuar**.

   O DCM determina automaticamente o formato do arquivo.
9. Se o certificado for um certificado **Servidor ou cliente**, digite a senha no quadro de tarefas e clique em **Continuar**. O DCM informará que o certificado foi importado.

Removendo certificados no IBM i

Use este procedimento para remover certificados pessoais.

Procedimento

1. Acesse a interface DCM, conforme descrito em "Acessando DCM" na página 280.
2. No painel de navegação, clique em **Selecionar um Armazenamento de Certificados**.

   A página Selecionar um Armazenamento de Certificados é exibida no quadro de tarefas.
3. Selecione a caixa de opção **Outro Armazenamento de Certificados do Sistema** e clique em **Continuar**.

   A página Armazenamento de Certificados e Senha é exibida.

4. No campo **Caminho e Nome de Arquivo do Armazenamento de Certificados**, digite o caminho e o nome de arquivo do IFS configurados durante a criação do armazenamento de certificados.
5. Digite uma senha no campo **Senha do Armazenamento de Certificados**. Clique em **Continue**.
   A página Armazenamento de Certificados Atual é exibida no quadro de tarefas.
6. Na categoria de tarefas **Gerenciar Certificados** no painel de navegação, clique em **Excluir Certificado**.
   A página Confirmar Exclusão de Certificado é exibida no quadro de tarefas.
7. Selecione o certificado que deseja excluir. Clique em **Excluir**.
8. Clique em **Sim** para confirmar que deseja excluir o certificado. Caso contrário, clique em **Não**.
   O DCM o informará se o certificado tiver sido excluído.

*Usando o armazenamento de certificados *SYSTEM para autenticação unidirecional no IBM i*

Siga estas instruções para configurar a autenticação unilateral.

Antes de começar

- Crie um gerenciador de filas, canais e filas de transmissão.
- Crie um certificado de servidor ou cliente no gerenciador de filas do servidor.
- Transfira o certificado de autoridade de certificação para o gerenciador de filas do cliente e importo-o no repositório de chaves.
- Inicie um listener nos gerenciadores de filas do servidor e do cliente.

Sobre esta tarefa

Para usar a autenticação unidirecional, usando um computador executando IBM i como o servidor TLS, configure o parâmetro SSL Key Repository (SSLKEYR) para *SYSTEM. Esta configuração registra o gerenciador de filas do IBM MQ como um aplicativo. É possível então designar um certificado ao gerenciador de filas para ativar uma autenticação unilateral.

Também é possível usar keystores privadas para implementar a autenticação unilateral criando um certificado simulado para o gerenciador de filas do cliente no repositório de chaves.

Procedimento

1. Execute as seguintes etapas nos gerenciadores de filas do servidor e do cliente:
   a) Altere o gerenciador de filas para configurar o parâmetro SSLKEYR, emitindo o comando `CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)`.
   b) Armazene em arquivo stash a senha do repositório de chaves padrão, emitindo o comando `CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')`.
      A senha deve estar entre aspas simples.
   c) Altere os canais para que tenham o CipherSpec correto no parâmetro SSLCIPHER.
   d) Atualize a segurança TLS emitindo o comando `RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)`.
2. Designe o certificado ao gerenciador de filas do servidor usando o DCM, conforme a seguir:
   a) Acesse a interface DCM, conforme descrito em "Acessando DCM" na página 280.
   b) No painel de navegação, clique em **Selecionar um Armazenamento de Certificados**.
      A página Selecionar um Armazenamento de Certificados é exibida no quadro de tarefas.
   c) Selecione o armazenamento de certificados *SYSTEM e clique em **Continuar**.
   d) No painel esquerdo, expanda **Gerenciar Aplicativos**.
   e) Selecione a definição **Visualização do aplicativo** para verificar se o gerenciador de filas foi registrado como um aplicativo.
      `SSL (WMQ)` está listado na tabela.

f) Selecione **Atualizar Designação de Certificado**.

g) Selecione **Servidor** e clique em **Continuar**.

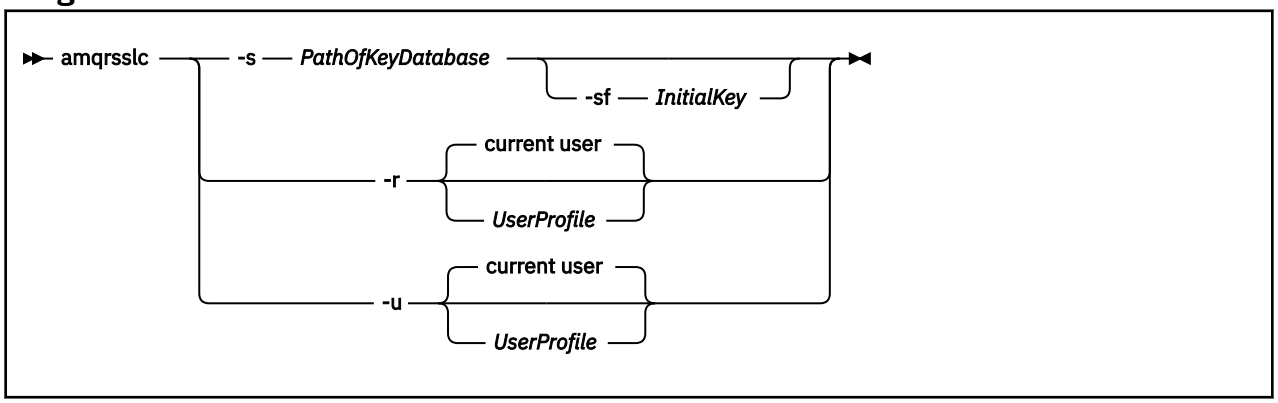
h) Selecione QMGRNAME (WMQ) e clique em **Atualizar Designação de Certificado**.

i) Selecione o certificado e clique em **Designar Novo Certificado**. Uma janela é aberta indicando que o certificado foi designado ao aplicativo.

Utilitário do cliente SSL (amqrsslc) do IBM MQ para IBM i

O utilitário do cliente SSL (amqrsslc) do IBM MQ para o IBM i é usado pelo IBM MQ MQI client nos sistemas IBM i para registrar ou cancelar registro do perfil do usuário cliente ou armazenar em arquivo stash a senha de armazenamento de certificados. O utilitário pode somente ser executado por um usuário com um perfil com a autoridade especial *ALLOBJ ou um membro do QMQMADM que tem opções de criar ou excluir registros de aplicativos no Digital Certificate Manager (DCM).

Diagrama de sintaxe



Registre o perfil do usuário do cliente

Se o IBM MQ MQI client estiver usando o armazenamento de certificados *SYSTEM, você deverá registrar o perfil do usuário cliente (usuário de logon) para uso como um aplicativo com Digital Certificate Manager (DCM).

Se você desejar registrar o perfil do usuário do cliente, execute o programa **amqrsslc** com a opção `-r` com o *UserProfile*. O perfil do usuário usado ao chamar **amqrsslc** deve ter autoridade *USE. Fornecer o *UserProfile* com a opção `-r` registra o *UserProfile* como um aplicativo do servidor com um rótulo do aplicativo exclusivo de QIBM_WEBSPHERE_MQ_*UserProfile* e um rótulo com uma descrição do *UserProfile* (WMQ). Esse aplicativo do servidor, em seguida, é exibido no DCM e é possível atribuir a este aplicativo qualquer servidor ou certificado cliente no armazenamento do sistema.

Nota: Se um perfil do usuário não for especificado com a opção `-r`, então, o perfil do usuário do usuário que estiver executando a ferramenta **amqrsslc** será registrado.

O código a seguir usa o **amqrsslc** para registrar um perfil do usuário. No primeiro exemplo, o perfil de usuário especificado está registrado; no segundo é o perfil do usuário com login efetuado:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

Cancelar o registro do perfil do usuário do cliente

Para cancelar o registro do perfil do cliente, execute o programa **amqrsslc** com a opção `-u` com *UserProfile*. O perfil do usuário usado ao chamar **amqrsslc** deve ter autoridade *USE. Fornecer o *UserProfile* com a opção `-u` cancela o registro do *UserProfile* com o rótulo QIBM_WEBSPHERE_MQ_*UserProfile* do DCM.

Nota: Se um perfil do usuário não for especificado com a opção `-u`, o perfil do usuário do usuário que estiver executando o **amqrsslc** ferramenta terá o registro cancelado.

O código a seguir usa **amqrsslc** para cancelar o registro de um perfil do usuário. No primeiro exemplo, o perfil de usuário especificado não está registrado; no segundo é o perfil do usuário com login efetuado:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

Armazenar em arquivo stash a senha de armazenamento de certificados

Se o IBM MQ MQI client não estiver usando o armazenamento de certificados *SYSTEM e usando outro armazenamento de certificados (ou seja, MQSSLKEYR for configurado para um valor diferente de *SYSTEM), a senha do banco de dados de chaves poderá ser armazenada em arquivo stash para que ele não precise ser especificado pelo aplicativo cliente quando for executado.

Use a opção `-s` para armazenar em stash a senha do banco de dados de chaves. Especifique o caminho completo e o nome do banco de dados de chave Se a extensão de arquivo não for fornecida, ela será considerada `.kdb`.

No código a seguir, o nome completo do arquivo do armazenamento de certificados é `/Path/Of/KeyDatabase/MyKey.kdb`:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

A execução deste código resulta em um pedido para a senha desse banco de dados de chaves. Essa senha é armazenada em um arquivo com o mesmo nome que o banco de dados de chaves com uma extensão `.sth`

Além disso, a chave inicial para criptografar a senha pode ser especificada A chave inicial deve ser armazenada em um arquivo como uma única linha de texto e, em seguida, o local desse arquivo é fornecido ao programa por meio da sinalização **-sf** . Se nenhum arquivo de chave inicial for fornecido, uma chave padrão será usada para criptografar a senha

O arquivo stash é armazenado no mesmo caminho que o banco de dados de chave. O exemplo de código gera um arquivo stash de `/Path/Of/KeyDatabase/MyKey.sth`.

O QMQM é o proprietário do usuário e o QMQMADM o proprietário do grupo para este arquivo. O QMQM e QMQMADM têm permissão de leitura, de gravação e outros perfis têm apenas permissão de leitura.

Quando mudanças nos certificados ou no armazenamento de certificados tornam-se efetivas no IBM i

Ao alterar os certificados em um armazenamento de certificados, ou o local do armazenamento de certificados, as mudanças entrarão em vigor, dependendo do tipo de canal e de como o canal está sendo executado.

Mudanças nos certificados no armazenamento de certificados e no atributo de repositório de chaves entrarão em vigor nas seguintes situações:

- Quando um novo processo de canal único de saída executa um canal do TLS pela primeira vez.
- Quando um novo processo de canal único TCP/IP de entrada recebe pela primeira vez uma solicitação para iniciar um canal do TLS.
- Quando o comando MQSC REFRESH SECURITY TYPE(SSL) é emitido para atualizar o ambiente TLS do IBM MQ.
- Para processos do aplicativo cliente, quando a última conexão do TLS no processo é fechada. A próxima conexão do TLS escolhe as mudanças de certificado.
- Para canais que são executados como encadeamentos de um processo de conjunto de processo (amqrmppa), quando o processo de conjunto de processo é iniciado ou reiniciado e executa pela primeira vez um canal do TLS. Se o processo de conjunto de processo já tiver executado um canal do

TLS e você desejar que a mudança entre em vigor imediatamente, execute o comando MQSC REFRESH SECURITY TYPE(SSL).

- Para canais que são executados como encadeamentos do inicializador de canal, quando o inicializador de canais é iniciado ou reiniciado e executa um canal do TLS pela primeira vez. Se o processo do inicializador de canais já tiver executado um canal do TLS e você desejar que a mudança entre em vigor imediatamente, execute o comando MQSC REFRESH SECURITY TYPE(SSL).
- Para canais que são executados como encadeamentos de um listener TCP/IP, quando o listener é iniciado ou reiniciado e recebe pela primeira vez uma solicitação para iniciar um canal do TLS. Se o listener já tiver executado um canal do TLS e você desejar que a mudança entre em vigor imediatamente, execute o comando MQSC REFRESH SECURITY TYPE(SSL).

Configurando hardware de criptografia no IBM i

Use este procedimento para configurar o Coprocessador Criptográfico no IBM i

Antes de começar

Verifique se o seu perfil de usuário possui autoridades especiais *ALLOBJ e *SECADM para que seja possível configurar o hardware do co-processador.

Procedimento

1. Acesse `http://`*`machine`*`.domain:2001` ou `https://`*`machine`*`.domain:2010`, em que *machine* é o nome do computador.

   Uma caixa de diálogo é exibida solicitando um nome de usuário e uma senha.
2. Digite um perfil válido do usuário e senha do IBM i.
3. Acesse Criptografia e siga os links apropriados para obter informações adicionais.

Como proceder a seguir

Para obter informações mais específicas sobre como configurar o Coprocessador Criptográfico 4767, veja Coprocessador Criptográfico 4767.

ALW Trabalhando com SSL/TLS no AIX, Linux, and Windows

Em sistemas AIX, Linux, and Windows, o suporte de Segurança da Camada de Transporte (TLS) é instalado com o IBM MQ.

Nota: Deprecated V 9.4.0 No IBM MQ 9.4.0, o uso de repositórios de chaves CMS e arquivos stash com aplicativos IBM MQ Java foi descontinuado. Migre para usar repositórios de chaves PKCS #12 e proteja as senhas do repositório de chaves usando o sistema de proteção de senha IBM MQ .

Importante: V 9.4.0 V 9.4.0 A partir do IBM MQ 9.4.0, os repositórios de chaves e arquivos stash do CMS não são suportados com canais AMQP e MQTT que usam SSL/TLS. Use repositórios de chaves PKCS #12 e proteja as senhas do repositório de chaves usando o sistema de proteção de senha IBM MQ .

Para obter informações mais detalhadas sobre as políticas de validação de certificado, consulte Validação do certificado e design da política de confiança.

Para obter mais informações sobre os comandos usados para gerenciar repositórios de chaves e certificados no AIX, Linux, and Windows, consulte “Comandos runmqakm e runmqktool em AIX, Linux, and Windows” na página 550.

ALW *Configurando um repositório de chaves no AIX, Linux, and Windows*

Siga este procedimento para criar um novo repositório de chaves

Antes de começar

Um repositório de chaves é protegido por uma senha, pois ele contém informações confidenciais Antes de criar o repositório de chaves, revise as opções fornecidas pelo IBM MQ para armazenar com segurança a senha do repositório de chaves. Para obter mais informações, consulte “Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301.

Nota: Deprecated V 9.4.0 No IBM MQ 9.4.0, o uso de repositórios de chaves CMS e arquivos stash com aplicativos IBM MQ Java foi descontinuado. Migre para usar repositórios de chaves PKCS #12 e proteja as senhas do repositório de chaves usando o sistema de proteção de senha IBM MQ .

Importante: V 9.4.0 V 9.4.0 A partir do IBM MQ 9.4.0, os repositórios de chaves e arquivos stash do CMS não são suportados com canais AMQP e MQTT que usam SSL/TLS. Use repositórios de chaves PKCS #12 e proteja as senhas do repositório de chaves usando o sistema de proteção de senha IBM MQ . É possível criar um repositório de chaves PKCS #12 usando o comando a seguir:

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

Esse comando cria um arquivo de repositório de chaves PKCS #12 denominado *filename*.p12 que é protegido com a senha especificada

Sobre esta tarefa

Uma conexão TLS requer um *repositório de chaves* em cada extremidade da conexão. Cada gerenciador de filas do IBM MQ e IBM MQ MQI client deverá ter acesso a um repositório de chaves. Para obter mais informações, consulte “O repositório de chaves SSL/TLS” na página 26.

Certificados digitais são armazenados no repositório de chaves. Esses certificados digitais têm rótulos. O rótulo do certificado associa um certificado pessoal a um gerenciador de fila específico ou IBM MQ MQI client O TLS usa esse certificado para propósitos de autenticação. Em AIX, Linux, and Windows sistemas, IBM MQ usa um dos valores a seguir para o rótulo certificado:

- O valor do gerenciador de filas ou atributo do canal **CERTLABL** , se ele estiver configurado.
- O valor padrão de `ibmwebspheremq`, com o nome do gerenciador de filas ou ID de logon do usuário IBM MQ MQI client anexado, tudo em letras minúsculas.

Para obter mais informações, consulte Etiquetas de certificado digital

O nome do arquivo do repositório de chaves inclui um caminho e um nome de raiz:

- Em sistemas AIX and Linux, o caminho padrão para um Gerenciador de Filas (configurado quando você criou o Gerenciador de Filas) é `/var/mqm/qmgrs/`*queue_manager_name*`/ssl`.

  Em sistemas Windows , o caminho padrão é *MQ_DATA_PATH*`\qmgrs\`*queue_manager_name*`\ssl`, em que *MQ_DATA_PATH* é o caminho de dados selecionado durante a instalação do IBM MQ. Por exemplo, `C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl`.

  O nome do arquivo padrão é key . kdb. Como alternativa, é possível usar seu caminho e nome do arquivo.

  Se você escolher seu próprio caminho ou nome de arquivo, configure as permissões para que o arquivo controle rigorosamente o acesso a ele.
- Para um cliente IBM MQ , não há nenhum caminho ou nome de arquivo padrão Controle rigorosamente o acesso a esse arquivo.

Não crie os repositórios de chaves em um sistema de arquivos que não suporta bloqueios no nível de arquivos, por exemplo, o NFS versão 2 em sistemas Linux.

Consulte “Alterando o local do repositório de chaves de um gerenciador de filas no AIX, Linux, and Windows” na página 305 para obter informações sobre a verificação e a especificação do nome do arquivo de banco de dados de chave. É possível especificar o nome do arquivo do banco de dados de chaves antes ou depois que o repositório de chaves é criado

É possível usar os comandos **runmqakm** (GSKCapiCmd) ou V 9.4.0 V 9.4.0 **runmqktool** (keytool) para gerenciar repositórios de chaves que são usados pelo IBM MQ. Para obter mais informações, consulte "Comandos runmqakm e runmqktool em AIX, Linux, and Windows" na página 550.

O ID do usuário que executa os comandos para gerenciar o repositório de chave deve ter permissão de gravação para o diretório no qual o arquivo do repositório de chave é criado ou atualizado... Para um gerenciador de filas que usa o diretório `ssl` padrão, o ID do usuário que executa o comando **runmqakm** ou **runmqktool** deve ser um membro do grupo mqm. Para um IBM MQ MQI client, se você executar **runmqakm** ou **runmqktool** a partir de um ID do usuário diferente do ID do usuário que executa o cliente, deverá alterar as permissões de arquivo para permitir que o IBM MQ MQI client acesse o repositório de chaves. Para obter mais informações, consulte "Acessando e protegendo seus arquivos do banco de dados de chaves no Windows" na página 303 ou "Acessando e Protegendo seus Arquivos do Banco de Dados de Chaves nos Sistemas AIX and Linux" na página 303.

É possível criar um novo repositório de chaves vazio usando o comando **runmqakm** .
V 9.4.0 V 9.4.0 Se você usar o comando **runmqktool** no lugar, o repositório de chaves será criado quando um comando for emitido para criar ou importar um certificado.

Nota: Se deve-se gerenciar certificados TLS de uma maneira que esteja de acordo com FIPS, use o comando **runmqakm**.

Procedimento

1. Emita o seguinte comando para criar um repositório de chaves com o comando **runmqakm** :

```
runmqakm -keydb -create -db filename -pw password -type type
         -stash -fips -strong
```

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

-type *tipo*
V 9.4.0 V 9.4.0 Especifica o tipo do repositório de teclas Para um repositório de chaves que é usado pelo IBM MQ, os valores possíveis são:

- `pkcs12`
- Deprecated `cms`

  Nota: No IBM MQ 9.4.0, o uso de repositórios de chaves e arquivos stash do CMS foi descontinuado para aplicativos IBM MQ Java e não é suportado para canais AMQP e MQTT que usam SSL/TLS.

-stash
Opcional. Especifique esta opção para armazenar a senha do repositório de chave em um arquivo stash Não será necessário armazenar a senha em um arquivo stash se você criptografar a senha usando o sistema de proteção de senha IBM MQ no lugar.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que são validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

-strong
Verifica se a senha inserida atende aos requisitos mínimos de força da senha. Os requisitos mínimos para uma senha são como a seguir:

- A senha deve ter no mínimo 14 caracteres.

- A senha deve conter no mínimo um caractere minúsculo, um caractere maiúsculo e um dígito ou caractere especial. Os caracteres especiais incluem o asterisco (*), o sinal de dólar ($), o sinal de número (#) e o sinal de percentual (%). Um espaço é classificado como um caractere especial.
- Cada caractere pode ocorrer no máximo de três vezes em uma senha.
- O máximo de dois caracteres consecutivos na senha podem ser idênticos.
- Todos os caracteres estão configurados no padrão para caracteres para impressão ASCII dentro do intervalo –. 0x20 - 0x7E.

2. Configure as permissões de acesso para os arquivos do repositório de chaves, conforme descrito em “Acessando e protegendo seus arquivos do banco de dados de chaves no Windows” na página 303 ou “Acessando e Protegendo seus Arquivos do Banco de Dados de Chaves nos Sistemas AIX and Linux” na página 303

   No Windows, por padrão, apenas o ID do usuário que executou o comando para criar o repositório de chaves recebe acesso para ler o arquivo stash (`.sth`). Após a criação de um arquivo stash com o comando **runmqakm** , verifique as permissões de arquivo e conceda permissão para a conta de serviço que está executando o gerenciador de filas ou para um grupo, como `mqmlocal`
3. Se você não estiver usando um arquivo stash, forneça a senha do keystore para o gerenciador de filas ou aplicativo cliente seguindo as instruções em “Fornecendo a senha do repositório de chaves para um gerenciador de filas no AIX, Linux, and Windows” na página 305 ou “Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on AIX, Linux, and Windows” na página 307.

Como proceder a seguir

Inclua certificados de autoridade de certificação (CA) padrão no repositório de chaves vazio, se necessário. Para obter informações adicionais, consulte “Incluindo certificados CA padrão em um repositório de chaves vazio no AIX, Linux, and Windows” na página 304.

ALW *Gerando senhas fortes para proteção do repositório de chaves no AIX, Linux, and Windows*

É possível gerar senhas fortes para o repositório de chaves de proteção usando o comando **runmqakm** (GSKCapiCmd).

É possível usar o comando **runmqakm** com os parâmetros a seguir para gerar uma senha forte:

```
runmqakm -random -create -length password_length -strong -fips
```

em que *password_length* é o comprimento da senha para gerar. O comprimento mínimo da senha que pode ser especificado é 14..

Ao usar a senha gerada no parâmetro **-pw** dos comandos de administração de certificados subsequentes, sempre coloque aspas duplas ao redor da senha. Nos sistemas AIX and Linux, também deve-se usar um caractere de barra invertida para escapar os caracteres a seguir se eles aparecerem na sequência de senha:

```
!  \  "  '
```

Ao inserir uma senha do repositório de chaves em resposta a um prompt a partir do comando **runmqakm** ou V 9.4.0 V 9.4.0 **runmqktool** , não é necessário citar ou escapar a senha, pois o shell do sistema operacional não afeta a entrada de dados nesses casos.

ALW *Criptografando senhas do repositório de chaves no AIX, Linux, and Windows*

Vários componentes do IBM MQ precisam de acesso a um repositório de chaves que contém certificados digitais ou chaves simétricas.. Um repositório de chaves é protegido por uma senha, pois ele contém informações confidenciais A senha do repositório de chaves deve ser armazenada em um local no qual o IBM MQ possa lê-lo quando o repositório de chaves for acessado A senha também deve ser criptografada para reduzir a probabilidade de acesso não autorizado para o repositório de chaves

Os seguintes componentes e recursos do IBM MQ suportam dois métodos diferentes para armazenar senhas do repositório de chaves:

- O repositório de chaves do TLS do gerenciador de fila
- IBM MQ MQI clients que usam o TLS
- V 9.4.0 A configuração de HA nativa na sub-rotina **NativeHALocalInstance** do arquivo `qm.ini`.
- V 9.4.0 A configuração de autenticação do token na sub-rotina **AuthToken** do arquivo `qm.ini`.

As senhas do repositório de chaves para uso por esses componentes podem ser criptografadas e armazenadas usando um dos métodos a seguir:

O sistema de proteção de senha do IBM MQ

Cada componente do IBM MQ fornece um comando para criptografar a senha do repositório de chaves O comando criptografado que o comando gera é armazenado em um arquivo.

Para o repositório de chaves TLS do gerenciador de filas, a senha é criptografada quando o atributo do gerenciador de filas **SSLKEYRPWD** é configurado

A senha é criptografada com o algoritmo AES-128 Os detalhes desse algoritmo são publicamente conhecidos e são considerados seguros.

A senha é armazenada em um formato proprietário que não é entendido por outro software que pode acessar o repositório de chaves.

Uma senha que é criptografada por um componente IBM MQ não pode ser usada por um componente IBM MQ diferente.

Uma chave de criptografia exclusiva pode ser fornecida quando a senha do repositório de chaves é criptografada Uma chave de criptografia exclusiva impede que qualquer pessoa que não tenha acesso à chave de criptografia possa decriptografar a senha.

A senha do repositório de chaves de texto simples é necessária para gerenciar os certificados que estão no repositório de chaves. Além de criptografar a senha do repositório de chaves usando o sistema de proteção de senha IBM MQ , você também deve armazenar a senha do repositório de chaves em um local seguro no qual ela possa ser acessada para esse propósito.

Para obter mais informações sobre o sistema de proteção de senha do IBM MQ , consulte “Protegendo senhas em arquivos de configuração do componente do IBM MQ” na página 574

Um arquivo stash do repositório de chave

O comando **runmqakm** pode armazenar a senha do repositório de chave em um arquivo stash..

A senha é criptografada com um método proprietário específico para o provedor criptográfico do IBM MQ, IBM Global Security Kit (GSKit).

Uma chave de criptografia exclusiva não pode ser fornecida

A senha criptografada é armazenada em um arquivo stash no mesmo diretório que o arquivo do repositório de chaves

Qualquer pessoa com acesso de leitura ao repositório de chaves e ao arquivo stash pode acessar e gerenciar o conteúdo do repositório de chaves..

Nota: Deprecated V 9.4.0 Em IBM MQ 9.4.0, o uso de arquivos stash com aplicativos IBM MQ Java foi descontinuado.

Importante: V 9.4.0 V 9.4.0 No IBM MQ 9.4.0, os arquivos stash não são suportados pelos canais AMQP e MQTT que usam TLS.

Independentemente do método escolhido para criptografar a senha do repositório de chaves, assegure-se de estar ciente das limitações da criptografia de senhas armazenadas. Para obter informações adicionais, consulte “Os limites para proteção por meio de criptografia de senha” na página 581.

Conceitos relacionados

“Fornecendo a senha do repositório de chaves para um gerenciador de filas no AIX, Linux, and Windows” na página 305
Como o repositório de chaves contém informações confidenciais, ele é protegido por uma senha Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

“Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on AIX, Linux, and Windows” na página 307
Como o repositório de chaves contém informações confidenciais, ele é protegido por uma senha Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

“Trabalhando com SSL/TLS no AIX, Linux, and Windows” na página 298
Em sistemas AIX, Linux, and Windows, o suporte de Segurança da Camada de Transporte (TLS) é instalado com o IBM MQ.

Windows *Acessando e protegendo seus arquivos do banco de dados de chaves no Windows*

Os arquivos do banco de dados de chave podem não ter as permissões de acesso apropriadas. Você deve configurar o acesso apropriado a esses arquivos.

Configure o controle de acesso para os arquivos *key*`.p12`, *key*`.kdb`, *key*`.sth`, *key*`.crle` *key*`.rdb`, em que *key* é o nome raiz de seu banco de dados de chaves para conceder autoridade a um conjunto restrito de usuários.

Se você tiver usado uma extensão de repositório de chaves diferente de `.p12` ou `.kdb`, também deverá assegurar que as permissões desse arquivo sejam configuradas.

Considere conceder acesso da seguinte forma:

autoridade plena
BUILTIN\Administrators, NT AUTHORITY\SYSTEM e o usuário que criou os arquivos de banco de dados.

autoridade de leitura
Para um gerenciador de filas, apenas o grupo mqm local. Isto supõe que o MCA esteja em execução com um ID do usuário no grupo mqm.

Para um cliente, o ID do usuário com o qual o processo do cliente está sendo executado.

Linux AIX *Acessando e Protegendo seus Arquivos do Banco de Dados de Chaves nos Sistemas AIX and Linux*

Os arquivos do banco de dados de chave podem não ter as permissões de acesso apropriadas. Você deve configurar o acesso apropriado a esses arquivos.

Para um gerenciador de filas, configure permissões nos arquivos de banco de dados de chave para que os processos de gerenciador de filas e de canal possam lê-los quando necessário, mas outros usuários não possam ler ou modificá-los. Normalmente, o usuário mqm precisa de permissões de leitura. Se você tiver criado o arquivo do banco de dados de chave efetuando login como o usuário mqm, as permissões serão provavelmente suficientes; se você não era o usuário mqm, mas outro usuário nesse grupo, provavelmente precisará conceder permissões de leitura a outros usuários no grupo mqm.

De forma semelhante para um cliente, configure permissões nos arquivos de banco de dados de chave para que os processos de cliente possam lê-los quando necessário, mas outros usuários não possam ler ou modificá-los. Normalmente, o usuário sob o qual o processo do cliente é executado precisa de permissões de leitura. Se você tiver criado o arquivo do banco de dados de chave efetuando login como esse usuário, as permissões serão provavelmente suficientes; se você não era o usuário do processo de cliente, mas outro usuário nesse grupo, provavelmente precisará conceder permissões de leitura a outros usuários no grupo.

Configure as permissões nos arquivos *key*`.p12`, *key*`.kdb`, *key*`.sth`, *key*`.crle` *key*`.rdb`, em que *key* é o nome raiz de seu banco de dados de chaves, para read e write para o proprietário do arquivo e para read para o grupo de usuários mqm ou cliente (-rw-r -----).

Se você tiver usado uma extensão de repositório de chaves diferente de `.p12` ou `.kdb`, também deverá assegurar que as permissões desse arquivo sejam configuradas.

ALW *Incluindo certificados CA padrão em um repositório de chaves vazio no AIX, Linux, and Windows*

Siga este procedimento para incluir um ou mais dos certificados de autoridade de certificação (CA) padrão em um repositório de chaves vazio...

Quando você cria um novo repositório de chaves, ele está vazio É possível incluir certificados de CA padrão em um repositório de chaves usando o comando **runmqakm** .

Usando o runmqakm

Emita o comando a seguir para incluir certificados de autoridade de certificação padrão em um repositório de chaves com o comando **runmqakm** :

```
runmqakm -cert -populate -db filename -pw password
```

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

Nota: O IBM MQ confia em todos os certificados que são assinados pelos certificados de CA em seu repositório de chaves. Considere cuidadosamente quais autoridades de certificação você deseja confiar e inclua apenas os certificados de CA que são necessários para autenticar seus clientes e gerenciadores de filas. Não é recomendado incluir o conjunto completo de certificados de CA padrão em um repositório de chaves.

ALW *Localizando o repositório de chaves para um gerenciador de filas no AIX, Linux, and Windows*

Use este procedimento para obter o local do arquivo do banco de dados de chave do gerenciador de filas

Procedimento

1. Exiba os atributos do seu gerenciador de filas, usando um dos seguintes comandos MQSC:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

   Também é possível exibir os atributos do gerenciador de filas usando o IBM MQ Explorer ou comandos PCF.
2. Examine a saída do comando para o nome do caminho e de raiz do arquivo de banco de dados da chave.
   Por exemplo,
   a. no AIX and Linux: `/var/mqm/qmgrs/QM1/ssl/key`, em que `/var/mqm/qmgrs/QM1/ssl` é o caminho e `key` e o nome derivado
   b. em Windows: *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl\key`, em que *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl` é o caminho e `key` é o nome da raiz O *MQ_INSTALLATION_PATH* representa o diretório de alto nível no qual o IBM MQ está instalado.

   Nota: No IBM MQ 9.3.0 , o campo SSLKEYR suporta um nome de arquivo completo (incluindo extensão) e um nome de raiz (sem extensão). Se um nome de raiz for configurado, IBM MQ anexará automaticamente `.kdb` e usará esse repositório de chaves.

ALW *Alterando o local do repositório de chaves de um gerenciador de filas no AIX, Linux, and Windows*

É possível alterar o local do arquivo de banco de dados de chave do gerenciador de filas por vários meios, incluindo o comando MQSC ALTER QMGR.

É possível alterar o local do arquivo de banco de dados de chave do seu gerenciador de filas, ao usar o comando MQSC ALTER QMGR para configurar o atributo do repositório de chaves do seu gerenciador de filas. Por exemplo, no AIX and Linux:

```
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/QM1/ssl/MyKey.kdb')
```

No Windows:

```
ALTER QMGR SSLKEYR('C:\Program Files\IBM\MQ\Qmgrs\QM1\ssl\Mykey.kdb')
```

Atenção: Em Windows e Linux, se os canais AMQP TLS forem usados, o sufixo do arquivo de repositório de chaves deverá ser um dos seguintes:

- `.kdb`, para um repositório de chaves CMS
- `.p12` ou `.pkcs12`para um repositório de chaves PKCS #12 .

Também é possível mudar os atributos do gerenciador de filas usando os comandos Explorer ou PCF do IBM MQ.

Ao alterar o local de um arquivo de banco de dados de chave do gerenciador de filas, os certificados não serão transferidos a partir do local antigo. Se o arquivo de banco de dados de chave que você está agora acessando for um novo arquivo de banco de dados de chave, você deverá preenchê-lo com os certificados pessoal e de autoridade de certificação de que precisa, conforme descrito em “Importando um certificado pessoal em um repositório de chaves no AIX, Linux, and Windows” na página 562.

Fornecendo a senha do repositório de chaves para um gerenciador de filas no AIX, Linux, and Windows

Como o repositório de chaves contém informações confidenciais, ele é protegido por uma senha Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

IBM MQ fornece dois mecanismos para fornecer a senha do repositório de chaves para um gerenciador de filas:

- “O atributo KEYRPWD” na página 305
- “O arquivo stash do repositório de chave” na página 306

Se você não usar um arquivo stash de repositório de chave, a senha do repositório de chaves será criptografada usando o sistema de proteção de senha IBM MQ Para obter mais informações sobre os métodos de proteção da senha do repositório de chaves, consulte “Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301

O atributo KEYRPWD

Para fornecer uma senha do repositório de chaves diretamente ao gerenciador de filas, execute o comando MQSC a seguir, substituindo *password* pela senha do repositório de chaves:

```
ALTER QMGR KEYRPWD('password')
```

Atenção: Assegure-se de colocar a senha entre aspas simples, caso contrário, o IBM MQ converterá os caracteres em maiúsculas.

Quando uma senha do repositório de chaves é especificada usando esse método, a senha é criptografada usando o sistema de proteção de senha do IBM MQ antes de ser armazenada

Uma chave de criptografia, conhecida como a chave inicial, é usada para criptografar a senha. Configure o gerenciador para usar uma chave inicial exclusiva para proteger com segurança a senha. Se você não fornecer uma chave inicial, a chave padrão será usada

Assegure que o gerenciador de filas esteja configurado com uma chave inicial exclusiva antes de configurar a senha do repositório de chaves. É possível modificar a chave inicial usando o atributo **INITKEY** no comando **ALTER QMGR** .. Por exemplo:

```
ALTER QMGR INITKEY('mykey')
```

Aviso: Modificar a chave inicial após configurar a senha do repositório de chaves não faz com que a senha do repositório de chaves seja criptografada com a nova chave inicial. A mudança da chave inicial sem também reconfigurar a senha do repositório de chaves resulta em IBM MQ não conseguir decriptografar a senha do repositório de chaves e, portanto, não conseguir acessar o repositório de chaves

Para obter mais informações sobre o atributos **KEYRPWD** , consulte KEYRPWD

O arquivo stash do repositório de chave

Se uma senha do repositório de chave não for fornecida para o gerenciador de filas usando o atributo **KEYRPWD** , IBM MQ assumirá que um arquivo stash existe no mesmo diretório que o repositório de chave. O arquivo stash tem o mesmo nome de raiz que o repositório de chaves, mas tem a extensão `.sth`

Um arquivo stash de repositório de chaves é criado ao mesmo tempo que o repositório de chaves, ou mais recente, como um comando **runmqakm** separado

Atenção: O formato do arquivo stash é específico do IBM MQ provedor criptográfico IBM Global Security Kit (GSKit)e não está disponível em plataformas que usam um provedor criptográfico diferente.

Para criar um arquivo stash quando o repositório de chaves for criado, especifique o parâmetro **-stash** Por exemplo:

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

onde *passw0rd* é a senha do repositório de chaves.

Para criar um arquivo stash posteriormente, execute o comando a seguir:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

onde *passw0rd* é a senha do repositório de chaves.

Conceitos relacionados

“Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301
Vários componentes do IBM MQ precisam de acesso a um repositório de chaves que contém certificados digitais ou chaves simétricas.. Um repositório de chaves é protegido por uma senha, pois ele contém informações confidenciais A senha do repositório de chaves deve ser armazenada em um local no qual o IBM MQ possa lê-lo quando o repositório de chaves for acessado A senha também deve ser criptografada para reduzir a probabilidade de acesso não autorizado para o repositório de chaves

“Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on AIX, Linux, and Windows” na página 307
Como o repositório de chaves contém informações confidenciais, ele é protegido por uma senha Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

ALW *Localizando o repositório de chaves para um IBM MQ MQI client no AIX, Linux, and Windows*

O local do repositório de chaves é fornecido pela variável MQSSLKEYR ou especificado na chamada MQCONNX.

Examine a variável de ambiente MQSSLKEYR para encontrar o local do arquivo de banco do dados de chave para seu IBM MQ MQI client. Por exemplo:

```
echo $MQSSLKEYR
```

Verifique também o aplicativo, porque o nome do arquivo do banco de dados de chave também poderá ser configurado em uma chamada MQCONNX, conforme descrito em “Especificando a o local do repositório de chaves para um IBM MQ MQI client no AIX, Linux, and Windows” na página 307. O valor definido em uma chamada MQCONNX substitui o valor de MQSSLKEYR.

ALW *Especificando a o local do repositório de chaves para um IBM MQ MQI client no AIX, Linux, and Windows*

Não há um repositório de chaves padrão para um IBM MQ MQI client. É possível especificar seu local em uma de duas maneiras. Certifique-se de que o arquivo do banco de dados de chaves somente possa ser acessado por usuários ou administradores pretendidos para evitar cópias não-autorizadas para outros sistemas.

É possível especificar o local do arquivo do banco de dados de chaves para o IBM MQ MQI client de duas maneiras:

- Configurar a variável de ambiente MQSSLKEYR. Por exemplo, no AIX and Linux:

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

  No Windows:

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

- Forneça o nome do caminho e de raiz do arquivo de banco de dados de chaves no campo *KeyRepository* da estrutura do MQSCO quando um aplicativo executar uma chamada MQCONNX. Para obter mais informações sobre a utilização da estrutura MQSCO em MQCONNX, consulte Visão geral para MQSCO.

Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on AIX, Linux, and Windows

Como o repositório de chaves contém informações confidenciais, ele é protegido por uma senha Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

O IBM MQ fornece quatro mecanismos para fornecer a senha do repositório de chaves para um IBM MQ MQI client:

- “Os campos KeyRepoPassword do MQSCO ” na página 307
- “A variável de ambiente MQKEYRPWD” na página 308
- “O atributo SSLKeyRepositoryPassword do arquivo de configuração do cliente” na página 308
- “O arquivo stash do repositório de chave” na página 309

Se você não usar um arquivo stash de repositório de chave, será possível fornecer a senha do repositório de chave como uma sequência de texto simples ou uma sequência que é criptografada usando o sistema de proteção de senha IBM MQ Para obter mais informações sobre os métodos de proteção da senha do repositório de chaves, consulte “Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301

Os campos KeyRepoPassword do MQSCO

Para fornecer uma senha do repositório de chaves usando a estrutura MQSCO, deve-se usar uma combinação dos três campos de sequência de variáveis a seguir:.

KeyRepoPasswordLength
O comprimento da senha.

KeyRepoPasswordPtr
Um ponteiro para o local na memória que contém a senha

KeyRepoPasswordOffset
O local da senha na memória, representado como o número de bytes do início da estrutura MQSCO.

Nota: É possível fornecer apenas um de **KeyRepoPasswordPtr** ou **KeyRepoPasswordOffset**

Por exemplo:

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

Atenção: Se você fornecer a senha usando esse método, criptografe a senha antes que ela seja fornecida para o aplicativo IBM MQ client Para obter mais informações, consulte “Criptografando a senha do repositório de chaves” na página 309.

Para obter mais informações sobre a estrutura MQCSO, consulte Opções de configuração de MQSCO-SSL/TLS

A variável de ambiente *MQKEYRPWD*

Se uma senha do repositório de chave não for fornecida ao cliente usando a estrutura MQSCO, será possível especificar a senha do repositório de chaves usando a variável de ambiente *MQKEYRPWD* . Por exemplo:

```
export MQKEYRPWD=passw0rd
```

ou

```
set MQKEYRPWD=passw0rd
```

em que `passw0rd` é a sua senha

Atenção: Se você fornecer a senha usando esse método, criptografe a senha antes de configurar o valor da variável de ambiente.. Para obter informações adicionais, consulte “Criptografando a senha do repositório de chaves” na página 309.

O atributo SSLKeyRepositoryPassword do arquivo de configuração do cliente

Se uma senha do repositório de chaves não for fornecida para o cliente usando um dos outros métodos, será possível especificar a senha do repositório de chaves usando o atributo **SSLKeyRepositoryPassword** na sub-rotina **SSL** do arquivo de configuração do cliente Por exemplo:

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

Atenção: Se você fornecer a senha usando esse método, criptografe a senha antes de configurar o valor do atributo **SSLKeyRepositoryPassword** .. Para obter mais informações, consulte “Criptografando a senha do repositório de chaves” na página 309.

Para obter mais informações sobre a sub-rotina SSL do arquivo de configuração do cliente, consulte Sub-rotina SSL do arquivo de configuração do cliente

O arquivo stash do repositório de chave

Se a senha do repositório de chave não for fornecida ao cliente usando um dos outros métodos, o IBM MQ assumirá que um arquivo stash existe no mesmo diretório que o repositório de chave. O arquivo stash tem o mesmo nome de raiz que o repositório de chaves, mas tem a extensão `.sth`

Um arquivo stash de repositório de chaves é criado ao mesmo tempo que o repositório de chaves, ou mais recente, usando um comando **runmqakm** separado

Atenção: O formato do arquivo stash é específico do IBM MQ provedor criptográfico IBM Global Security Kit (GSKit)e não está disponível em plataformas que usam um provedor criptográfico diferente.

Para criar um arquivo stash quando o repositório de chaves for criado, especifique o parâmetro **-stash** Por exemplo:

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

onde *passw0rd* é a senha do repositório de chaves.

Para criar um arquivo stash posteriormente, execute o comando a seguir:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

onde *passw0rd* é a senha do repositório de chaves.

Criptografando a senha do repositório de chaves

Se você fornecer a senha do repositório de chaves usando qualquer método diferente de um arquivo stash, criptografe a senha usando o sistema de proteção de senha IBM MQ . Para criptografar a senha, execute o comando **runmqicred** .. Insira a senha do repositório de chaves quando solicitado.. O comando gera a senha criptografada.. A senha criptografada pode ser fornecida para o IBM MQ MQI client em vez da senha de texto simples usando qualquer um dos métodos descritos.

Uma chave de criptografia, conhecida como a chave inicial, é usada para criptografar a senha. Quando você criptografar a senha, use uma chave inicial exclusiva para proteger com segurança a senha Para fornecer sua própria chave inicial, use o parâmetro **-sf** no comando **runmqicred** . Se você não fornecer uma chave inicial, a chave padrão será usada

Para obter mais informações, consulte runmqicred (proteger as senhas do cliente IBM MQ).

Se você fornecer a sua própria chave inicial quando a senha do repositório de chaves for criptografada e fornecer a senha criptografada para o IBM MQ MQI client, também deverá assegurar que você forneça a mesma chave inicial para o IBM MQ MQI client Para obter mais informações sobre como fornecer a chave inicial para um IBM MQ MQI client, consulte “Fornecendo uma chave inicial para um IBM MQ MQI client em AIX, Linux, and Windows” na página 310

Conceitos relacionados

“Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301
Vários componentes do IBM MQ precisam de acesso a um repositório de chaves que contém certificados digitais ou chaves simétricas.. Um repositório de chaves é protegido por uma senha, pois ele contém informações confidenciais A senha do repositório de chaves deve ser armazenada em um local no qual o IBM MQ possa lê-lo quando o repositório de chaves for acessado A senha também deve ser criptografada para reduzir a probabilidade de acesso não autorizado para o repositório de chaves

“Fornecendo a senha do repositório de chaves para um gerenciador de filas no AIX, Linux, and Windows” na página 305
Como o repositório de chaves contém informações confidenciais, ele é protegido por uma senha Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

ALW *Fornecendo uma chave inicial para um IBM MQ MQI client em AIX, Linux, and Windows*
Se você fornecer variáveis para um IBM MQ MQI client que tenha sido criptografado usando o IBM MQ Password Protection System, poderá ser necessário fornecer a chave inicial correspondente que foi usada para criptografar o valor

Se você não especificou uma chave inicial ao criptografar o valor, não será necessário fornecer nenhum valor de chave inicial para o IBM MQ client.. No entanto, se você usou uma chave inicial exclusiva, será possível fornecer a chave inicial para o IBM MQ client usando os métodos a seguir:

- "Fornecendo a chave inicial usando a estrutura MQCSP" na página 310
- "Fornecendo a chave inicial usando a variável de ambiente MQS_MQI_KEYFILE" na página 310
- "Fornecendo a chave inicial usando o arquivo de configuração do cliente" na página 310

Fornecendo a chave inicial usando a estrutura MQCSP

Para fornecer a chave inicial usando a estrutura MQCSP, deve-se usar uma combinação dos três campos de sequência de variáveis a seguir:

InitialKeyLength
O comprimento da chave inicial

InitialKeyPtr
Um ponteiro para o local na memória contendo a chave inicial

InitialKeyOffset
O local da chave inicial na memória, representado como o número de bytes do início da estrutura MQCSP.

Nota: É possível fornecer apenas um de **InitialKeyPtr** ou **InitialKeyOffset**

Por exemplo:

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

Fornecendo a chave inicial usando a variável de ambiente MQS_MQI_KEYFILE

Se uma chave inicial não for fornecida ao cliente usando a estrutura MQCSP, IBM MQ verificará a variável de ambiente *MQS_MQI_KEYFILE* . Você deve configurar essa variável de ambiente para o local de um arquivo contendo uma única linha de texto, consistindo na chave inicial que deseja usar.

Por exemplo, se um arquivo chamado `mykey.key` existir no diretório-raiz e contiver a chave inicial, você deverá configurar a variável de ambiente da seguinte forma:

```
export MQS_MQI_KEYFILE=/mykey.key
```

ou

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

Fornecendo a chave inicial usando o arquivo de configuração do cliente

Se uma chave inicial não for fornecida ao cliente usando um mecanismo anterior, IBM MQ verificará o atributo **MQIInitialKeyFile** da sub-rotina de Segurança do arquivo `mqclient.ini` . Você deve configurar esse atributo para o local de um arquivo contendo uma única linha de texto, consistindo na chave inicial que deseja usar.

Por exemplo, se um arquivo chamado `mykey.key` existir no diretório raiz e contiver a chave inicial, o arquivo de configuração do cliente deverá conter o seguinte:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

Conceitos relacionados

"Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on AIX, Linux, and Windows" na página 307

Como o repositório de chaves contém informações confidenciais, ele é protegido por uma senha Para poder acessar o conteúdo do repositório de chaves para executar operações TLS, o IBM MQ deve ser capaz de recuperar a senha do repositório de chaves.

"Trabalhando com SSL/TLS" na página 280

Estes tópicos fornecem instruções para executar tarefas únicas relacionadas ao uso do TLS com o IBM MQ.

ALW *Quando as mudanças nos certificados ou no repositório de chaves tornam-se efetivas no AIX, Linux, and Windows*

Ao mudar os certificados em um repositório de chaves ou o local do repositório de chaves, as mudanças entram em vigor em um momento que depende do tipo de canal e de como o canal está em execução.

As mudanças nos certificados no repositório de chaves ou no local do repositório de chaves se tornam efetivas nas situações a seguir:

- Quando um novo processo de canal único de saída executa um canal do TLS pela primeira vez.
- Quando um novo processo de canal único TCP/IP de entrada recebe pela primeira vez uma solicitação para iniciar um canal do TLS.
- Quando o comando MQSC **REFRESH SECURITY TYPE(SSL)** é emitido para atualizar o ambiente TLS.
- Para processos do aplicativo cliente, quando a última conexão do TLS no processo é fechada. A próxima conexão do TLS selecionará as mudanças do certificado.
- Para canais que são executados como encadeamentos de um processo de conjunto de processo (amqrmppa), quando o processo de conjunto de processo é iniciado ou reiniciado e executa pela primeira vez um canal do TLS. Se o processo de conjunto de processos já tiver executado um canal TLS e você desejar que a mudança se torne efetiva imediatamente, execute o comando MQSC **REFRESH SECURITY TYPE(SSL)**.
- Para canais que são executados como encadeamentos do inicializador de canal, quando o inicializador de canais é iniciado ou reiniciado e executa um canal do TLS pela primeira vez. Se o processo do inicializador de canais já tiver executado um canal TLS e você desejar que a mudança se torne efetiva imediatamente, execute o comando **REFRESH SECURITY TYPE(SSL)** do MQSC.
- Para canais que são executados como encadeamentos de um listener TCP/IP, quando o listener é iniciado ou reiniciado e recebe pela primeira vez uma solicitação para iniciar um canal do TLS. Se o listener já tiver executado um canal TLS e você desejar que a mudança entre em vigor imediatamente, execute o comando MQSC **REFRESH SECURITY TYPE(SSL)**.

Também é possível atualizar o ambiente TLS do IBM MQ usando os comandos IBM MQ Explorer ou PCF..

Importante: As mudanças no arquivo de configuração do keystore ou no keystore que é usado por um interceptor do MCA Advanced Message Security (AMS) ou um cliente AMS entram em vigor quando o gerenciador de filas ou aplicativo é reiniciado.

ALW *Configurando para o hardware de criptografia no AIX, Linux, and Windows*

É possível configurar o hardware de criptografia para um gerenciador de filas ou cliente de várias maneiras.

É possível configurar o hardware de criptografia para um gerenciador de filas no AIX, Linux, and Windows usando um dos métodos a seguir:

- Use o comando MQSC do **ALTER QMGR** com o parâmetro SSLCRYP , conforme descrito em ALTER QMGR
- Use IBM MQ Explorer para configurar o hardware de criptografia em seu sistema AIX, Linux, and Windows . Para obter mais informações, consulte a ajuda online.

É possível configurar o hardware criptográfico para um cliente IBM MQ no AIX, Linux, and Windows usando um dos métodos a seguir:

- Configure a variável de ambiente **MQSSLCRYP** . Os valores permitidos para **MQSSLCRYP** são iguais aos do parâmetro **SSLCRYP** , conforme descrito em ALTER QMGR. Para configurar essa variável de ambiente, use um destes comandos:
  - Linux AIX Nos sistemas AIX and Linux:

    ```
    export MQSSLCRYP=string
    ```

  - Windows Nos sistemas Windows:

    ```
    SET MQSSLCRYP=string
    ```

  em que *string* representa a sequência de parâmetros a ser usada para configurar o hardware criptográfico presente no sistema

  Se você usar a versão GSK_PKCS11 do parâmetro **SSLCRYP** , o rótulo do token PKCS #11 deverá corresponder ao rótulo com o qual seu hardware foi configurado.
- Configure o atributo **SSLCryptoHardware** na sub-rotina SSL do arquivo de configuração IBM MQ client .. Os valores permitidos são iguais aos do parâmetro **SSLCRYP** , conforme descrito em **ALTER QMGR**.

  Se você usar a versão GSK_PKCS11 do parâmetro **SSLCRYP** , o rótulo do token PKCS #11 deverá corresponder ao rótulo com o qual seu hardware foi configurado.
- Configure o campo **CryptoHardware** da estrutura de opções de configuração de SSL, MQSCO, em uma chamada MQCONNX. Para obter mais informações, consulte Visão geral para MQSCO.

Atenção: >Ao fornecer a configuração para o hardware criptográfico por meio da variável de ambiente **MQSSLCRYP** ou do atributo **SSLCryptoHardware** , é necessário proteger a senha antes do armazenamento Para obter mais informações, consulte “IBM MQ clients que usam hardware criptográfico” na página 578.

Se você configurou um hardware criptográfico que usa a interface PKCS #11 usando qualquer um desses métodos, será necessário armazenar o certificado pessoal para ser usado nos canais no arquivo de banco de dados de chave para o token de criptografia que você configurou. Isso é descrito no “Gerenciando certificados em hardware PKCS #11” na página 571.

MQ Appliance Trabalhando com SSL/TLS no IBM MQ Appliance

O IBM MQ Appliance tem suporte a Segurança da Camada de Transporte (TLS).

O IBM MQ Appliance possui comandos distintos para o gerenciamento de certificados. Para obter informações detalhadas sobre gerenciamento de certificado, consulte a documentação do IBM MQ Appliance, Gerenciamento de certificado de TLS

z/OS Working with SSL/TLS on z/OS

This information describes how you set up and work with Transport Layer Security (TLS) on z/OS.

Each topic includes examples of performing each task using RACF. You can perform similar tasks using the other external security managers.

On z/OS, you must also set the number of server subtasks that each queue manager uses for processing TLS calls, as described in “Setting the SSLTASKS parameter on z/OS” on page 313.

z/OS TLS support is integral to the operating system, and is known as *System SSL*. System SSL is part of the Cryptographic Services Base element of z/OS. The Cryptographic Services Base members are installed in the *pdsname*. SIEALNKE partitioned data set (PDS). When you install System SSL, ensure that you choose the appropriate options to provide the CipherSpecs that you require.

If you need to renew a self-signed certificate, see Steps for renewing a self-signed certificate in RACF for more information.

z/OS *Requisitos adicionais de ID do usuário para TLS no z/OS*

Estas informações descrevem os requisitos adicionais que seu ID do usuário precisa para configurar e trabalhar com TLS no z/OS.

Assegure-se de que tenha todas as atualizações apropriadas de High Impact or Pervasive (HIPER) no sistema.

Se o repositório de chaves pertencer ao ID do usuário CHINIT, esse ID do usuário precisará de acesso de leitura ao IRR IRR.DIGTCERT.LISTRING na classe FACILITY, caso contrário, atualize o acesso e leia o acesso ao IRR.DIGTCERT.LIST perfil. Conceder acesso usando o comando PERMIT com ACCESS (UPDATE) ou ACCESS (READ) conforme apropriado.

Assegure-se de que tenha configurado os seguintes pré-requisitos:

- O ID do usuário *ssidCHIN* é definido corretamente no RACFe o ID do usuário *ssidCHIN* tem o acesso apropriado aos perfis a seguir.
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

  Essas variáveis são definidas na classe FACILITY do RACF.
- O ID do usuário *ssidCHIN* é o proprietário do conjunto de chave.
- O certificado pessoal do gerenciador de filas, se criado pelo comando RACDCERT, é criado com um ID do usuário de tipo de certificado que também é o mesmo que o ID do usuário *ssidCHIN*.
- O inicializador de canais é reciclado ou o comando **REFRESH SECURITY TYPE(SSL)** é emitido, para selecionar as mudanças feitas no conjunto de chaves.
- O procedimento de inicializador de canais do IBM MQ tem acesso a biblioteca de tempo de execução do SSL do Sistema *pdsname*.SIEALNKE por meio da lista de links, LPA ou uma instrução DD STEPLIB. Essa biblioteca deve ser autorizada pelo APF.
- O ID do usuário sob cuja autoridade o inicializador de canais está em execução está configurado para usar z/OS UNIX System Services (z/OS UNIX), conforme descrito na documentação do z/OS UNIX System Services Planning .

  Os usuários que não desejam que o inicializador de canais chame o z/OS UNIX usando o segmento de UID e OMVS do guest/padrão, precisam apenas modelar um novo segmento de OMVS com base no segmento padrão, já que o inicializador de canais não requer permissões especiais e não é executado dentro do UNIX como um superusuário.

  Consulte os comandos PERMIT no “Giving the channel initiator the correct access rights on z/OS” na página 315 para obter alguns exemplos de como você fornece ao inicializador de canais o acesso correto

z/OS *Setting the SSLTASKS parameter on z/OS*

Use the ALTER QMGR command to set the number of server subtasks for processing TLS calls

To use TLS channels, ensure that there are at least two server subtasks by setting the SSLTASKS parameter, using the ALTER QMGR command. For example:

```
ALTER QMGR SSLTASKS(5)
```

To avoid problems with storage allocation, do not set the SSLTASKS attribute to a value greater than eight in an environment where there is no Certificate Revocation List (CRL) checking.

If CRL checking is used, an SSLTASK is held by the channel concerned for the duration of that check. This could be for a significant elapsed time while the relevant LDAP server is contacted, because each SSLTASK is a z/OS task control block.

You must restart the channel initiator if you change the value of the SSLTASKS attribute.

z/OS *Setting up a key repository on z/OS*

Set up a key repository at both ends of the connection. Associate each key repository with its queue manager.

A TLS connection requires a *key repository* at each end of the connection. Each queue manager must have access to a key repository. Use the SSLKEYR parameter on the ALTER QMGR command to associate a key repository with a queue manager. See “O repositório de chaves SSL/TLS” on page 26 for more information.

On z/OS, digital certificates are stored in a *key ring* that is managed by your External Security Manager (ESM) . These digital certificates have labels, which associate the certificate with a queue manager. TLS uses these certificates for authentication purposes. All the examples that follow use RACF commands. Equivalent commands exist for other ESM programs.

On z/OS, IBM MQ uses either the value of the **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager appended. See Digital certificate labels for details.

The key repository name for a queue manager is the name of a key ring in your RACF database. You can specify the key ring name either before or after creating the key ring.

Use the following procedure to create a new key ring for a queue manager:

1. Ensure that you have the appropriate authority to issue the RACDCERT command (see Controlling the use of the RACDCERT command for more details).
2. Issue the following command:

   ```
   RACDCERT ID( userid1 ) ADDRING( ring-name )
   ```

   where:

   - *userid1* is the user ID of the channel initiator address space, or the user ID that is going to own the key ring (if the key ring is shared).
   - *ring-name* is the name you want to give to your key ring. The length of this name can be up to 237 characters. This name is case-sensitive. Specify *ring-name* in uppercase characters to avoid problems.

z/OS *Making CA certificates available to a queue manager on z/OS*

After you have created your key ring, connect any relevant CA certificates to it.

If you have the CA certificate in a data set, you must first add the certificate to the RACF database by using the following command:

```
RACDCERT ID( userid1 ) ADD( input-data-set-name ) WITHLABEL( 'My CA' )
```

Then to connect a CA certificate for My CA to your key ring, use the following command:

```
RACDCERT ID(userid1)
CONNECT(CERTAUTH LABEL('My CA') RING(ring-name) USAGE(CERTAUTH))
```

where *userid1* is either the channel initiator user ID or the owner of a shared key ring.

For more information about CA certificates, refer to “Certificados Digitais” on page 13.

z/OS *Locating the key repository for a queue manager on z/OS*

Use this procedure to obtain the location of your queue manager's key ring.

1. Display your queue manager's attributes, using either of the following MQSC commands:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

2. Examine the command output for the location of the key ring.

z/OS *Specifying the key repository location for a queue manager on z/OS*

To specify the location of your queue manager's key ring, use the ALTER QMGR MQSC command to set your queue manager's key repository attribute.

For example:

```
ALTER QMGR SSLKEYR(CSQ1RING)
```

if the key ring is owned by the channel initiator address space, or:

```
ALTER QMGR SSLKEYR(userid1/CSQ1RING)
```

if it is a shared key ring, where *userid1* is the user ID that owns the key ring.

z/OS *Giving the channel initiator the correct access rights on z/OS*

The channel initiator (CHINIT) needs access to the key repository and to certain security profiles.

Granting the CHINIT access to read the key repository

If the key repository is owned by the CHINIT user ID, this user ID needs read access to the IRR.DIGTCERT.LISTRING profile in the FACILITY class, and update access otherwise, and read access to the IRR.DIGTCERT.LIST profile. Grant access by using the PERMIT command with ACCESS(UPDATE) or ACCESS(READ) as appropriate:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID( userid ) ACCESS(UPDATE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID( userid ) ACCESS(READ)
```

where *userid* is the user ID of the channel initiator address space.

Granting the CHINIT read access to the appropriate CSF* profiles

For hardware support provided through the Integrated Cryptographic Service Facility (ICSF) to be used, ensure your CHINIT user ID has read access to the appropriate CSF* profiles in the CSFSERV class by using the following command:

```
PERMIT csf-resource CLASS(CSFSERV) ID( userid ) ACCESS(READ)
```

where *csf-resource* is the name of the CSF* profile and *userid* is the user ID of the channel initiator address space.

Repeat this command for each of the following CSF* profiles:

- CSFDSG
- CSFDSV
- CSFPKD
- CSFPKE
- CSFPKI

Your CHINIT user ID might also need read access to other CSF* profiles. For example, if you are using the ECDHE_RSA_AES_256_GCM_SHA384 Cipher Spec, your CHINIT user ID also needs read access to the following CSF* profiles:

- CSF1DVK
- CSF1GAV
- CSF1GKP
- CSF1SKE
- CSF1TRC
- CSF1TRD

For more information, see RACF CSFSERV resource requirements.

If your certificate keys are stored in ICSF and your installation has established access control over keys stored in ICSF, ensure your CHINIT user ID has read access to the profile in the CSFKEYS class by using the following command:

```
PERMIT IRR.DIGTCERT. userid.* CLASS(CSFKEYS) ID( userid ) ACCESS(READ)
```

where *userid* is the user ID of the channel initiator address space.

Using the Integrated Cryptographic Service Facility (ICSF)

The channel initiator can use ICSF to generate a random number when seeding the password protection algorithm to obfuscate passwords flowing over client channels if TLS is not being used.

For further information, see “Using the Integrated Cryptographic Service Facility (ICSF)” on page 267

z/OS *When changes to certificates or the key repository become effective on z/OS*

Changes become effective when the channel initiator starts or the repository is refreshed.

Specifically, changes to the certificates in the key ring and to the key repository attribute become effective on either of the following occasions:

- When the channel initiator is started or restarted.
- When the REFRESH SECURITY TYPE(SSL) command is issued to refresh the contents of the key repository.

z/OS *Creating a self-signed personal certificate on z/OS*

Use this procedure to create a self-signed personal certificate.

1. Generate a certificate and a public and private key pair using the following command:

```
RACDCERT ID(userid2) GENCERT
SUBJECTSDN(CN('common-name')
           T('title')
           OU('organizational-unit')
           O('organization')
```

```
          L('locality')
          SP('state-or-province')
          C('country'))
WITHLABEL('label-name')
```

2. Connect the certificate to your key ring using the following command:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.
- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.

  userid1 and *userid2* can be the same ID.
- *ring-name* is the name you gave the key ring in "Setting up a key repository on z/OS" on page 314.
- *label-name* must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager appended. See Digital certificate labels for details.

z/OS *Requesting a personal certificate on z/OS*

Apply for a personal certificate using RACF.

To apply for a personal certificate, use RACF as follows:

1. Create a self-signed personal certificate, as in "Creating a self-signed personal certificate on z/OS" on page 316. This certificate provides the request with the attribute values for the Distinguished Name.
2. Create a PKCS #10 Base64-encoded certificate request written to a data set, using the following command:

```
RACDCERT ID(userid2) GENREQ(LABEL(' label_name ')) DSN(' output_data_set_name ')
```

   where

   - *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space
   - *label_name* is the label used when creating the self-signed certificate

   See "Rótulos de Certificados Digitais, Entendendo os Requisitos" on page 27 for details.
3. Send the data set to a Certificate Authority (CA) to request a new personal certificate.
4. When the signed certificate is returned to you by the Certificate Authority, add the certificate back into the RACF database, using the original label, as described in "Adding personal certificates to a key repository on z/OS" on page 318.

z/OS *Creating a RACF signed personal certificate*

RACF can function as a certificate authority and issue its own CA certificate.

This section uses the term *signer certificate* to denote a CA certificate issued by RACF.

The private key for the signer certificate must be in the RACF database before you carry out the following procedure:

1. Use the following command to generate a personal certificate signed by RACF, using the signer certificate contained in your RACF database:

```
RACDCERT ID(userid2) GENCERT
SUBJECTSDN(CN('common-name')
          T('title')
          OU('organizational-unit')
```

```
           O('organization')
           L('locality')
           SP('state-or-province')
           C('country'))
WITHLABEL('label-name')
SIGNWITH(CERTAUTH LABEL('signer-label'))
```

2. Connect the certificate to your key ring using the following command:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.
- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.

  userid1 and *userid2* can be the same ID.
- *ring-name* is the name you gave the key ring in "Setting up a key repository on z/OS" on page 314.
- *label-name* must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details.
- *signer-label* is the label of your own signer certificate.

z/OS *Adding personal certificates to a key repository on z/OS*

Use this procedure to add or import a personal certificate to a key ring.

After the certificate authority sends you a new personal certificate, add it to the key ring using the following procedure:

1. Add the certificate to the RACF database using the following command:

```
RACDCERT ID( userid2 ) ADD( input-data-set-name ) WITHLABEL(' label-name ')
```

2. Connect the certificate to your key ring using the following command:

```
RACDCERT ID( userid1 )
CONNECT(ID( userid2 ) LABEL(' label-name ') RING( ring-name ) USAGE(PERSONAL))
```

where:

- *userid1* is the user ID of the channel initiator address space or owner of the shared key ring.
- *userid2* is the user ID associated with the certificate and must be the user ID of the channel initiator address space.
- *ring-name* is the name you gave the key ring in "Setting up a key repository on z/OS" on page 314.
- *input-data-set-name* is the name of the data set containing the CA signed certificate. The data set must be cataloged and must not be a PDS or a member of a PDS. The record format (RECFM) expected by RACDCERT is VB. RACDCERT dynamically allocates and opens the data set, and reads the certificate from it as binary data.
- *label-name* is the label name that was used when you created the original request. It must be either the value of the IBM MQ **CERTLABL** attribute, if it is set, or the default `ibmWebSphereMQ` with the name of the queue manager or queue sharing group appended. See Digital certificate labels for details.

z/OS *Exporting a personal certificate from a key repository on z/OS*

Export the certificate using the RACDCERT command.

On the system from which you want to export the certificate, use the following command:

```
RACDCERT ID(userid2) EXPORT(LABEL('label-name'))
DSN(output-data-set-name) FORMAT(CERTB64)
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the label of the certificate you want to extract.
- *output-data-set-name* is the data set into which the certificate is placed.
- CERTB64 is a DER encoded X.509 certificate that is in Base64 format. You can choose an alternative format, for example:

  CERTDER
  DER encoded X.509 certificate in binary format

  PKCS12B64
  PKCS #12 certificate in Base64 format

  PKCS12DER
  PKCS #12 certificate in binary format

z/OS *Deleting a personal certificate from a key repository on z/OS*

Delete a personal certificate using the RACDCERT command.

Before deleting a personal certificate, you might want to save a copy of it. To copy your personal certificate to a data set before deleting it, follow the procedure in "Exporting a personal certificate from a key repository on z/OS" on page 318. Then use the following command to delete your personal certificate:

```
RACDCERT ID( userid2 ) DELETE(LABEL(' label-name '))
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the name of the certificate you want to delete.

z/OS *Renaming a personal certificate in a key repository on z/OS*

Rename a certificate using the RACDCERT command.

If you do not want a certificate with a specific label to be found, but do not want to delete it, you can rename it temporarily using the following command:

```
RACDCERT ID( userid2 ) LABEL(' label-name ') NEWLABEL(' new-label-name ')
```

where:

- *userid2* is the user ID under which the certificate was added to the key ring.
- *label-name* is the name of the certificate you want to rename.
- *new-label-name* is the new name of the certificate.

This can be useful when testing TLS client authentication.

z/OS *Associating a user ID with a digital certificate on z/OS*

IBM MQ can use a user ID associated with a RACF certificate as a channel user ID. Associate a user ID with a certificate by installing it under that user ID, or using a Certificate Name Filter.

The method described in this topic is an alternative to the platform-independent method for associating a user ID with a digital certificate, which uses channel authentication records. For more information about channel authentication records, see "Registros de Autenticação de Canal" on page 53.

When an entity at one end of a TLS channel receives a certificate from a remote connection, the entity asks RACF if there is a user ID associated with that certificate. The entity uses that user ID as the channel user ID. If there is no user ID associated with the certificate, the entity uses the user ID under which the channel initiator is running.

Associate a user ID with a certificate in either of the following ways:

- Install that certificate into the RACF database under the user ID with which you want to associate it, as described in "Adding personal certificates to a key repository on z/OS" on page 318.
- Use a Certificate Name Filter (CNF) to map the Distinguished Name of the subject or issuer of the certificate to the user ID, as described in "Setting up a certificate name filter on z/OS" on page 320.

z/OS *Setting up a certificate name filter on z/OS*

Use the RACDCERT command to define a certificate name filter (CNF), which maps a Distinguished Name to a user ID.

Perform the following steps to set up a CNF.

1. Enable CNF functions using the following command. You require update authority on the class DIGTNMAP to do this.

```
SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
```

2. Define the CNF. For example:

```
RACDCERT ID(USER1) MAP WITHLABEL('filter1') TRUST
SDNFILTER('O=IBM.C=UK') IDNFILTER('O=ExampleCA.L=Internet')
```

   where USER1 is the user ID to be used when:

   - The DN of the subject has an Organization of `IBM` and a Country of UK.
   - The DN of the issuer has an Organization of `ExampleCA` and a Locality of `Internet`.

3. Refresh the CNF mappings:

```
SETROPTS RACLIST(DIGTNMAP) REFRESH
```

Note:

1. If the actual certificate is stored in the RACF database, the user ID under which it is installed is used in preference to the user ID associated with any CNF. If the certificate is not stored in the RACF database, the user ID associated with the most specific matching CNF is used. Matches of the subject DN are considered more specific than matches of the issuer DN.
2. Changes to CNFs do not apply until you refresh the CNF mappings.
3. A DN matches the DN filter in a CNF only if the DN filter is identical to the *least significant portion* of the DN. The least significant portion of the DN comprises the attributes that are usually listed at the right-most end of the DN, but which appear at the beginning of the certificate.

   For example, consider the SDNFILTER `'O=IBM.C=UK'`. A subject DN of `'CN=QM1.O=IBM.C=UK'` matches that filter, but a subject DN of `'CN=QM1.O=IBM.L=Hursley.C=UK'` does not match that filter.

   The least significant portion of some certificates can contain fields that do not match the DN filter. Consider excluding these certificates by specifying a DN pattern in the SSLPEER pattern on the DEFINE CHANNEL command.
4. If the most specific matching CNF is defined to RACF as NOTRUST, the entity uses the user ID under which the channel initiator is running.
5. RACF uses the `'.'` character as a separator. IBM MQ uses either a comma or a semicolon.

You can define CNFs to ensure that the entity never sets the channel user ID to the default, which is the user ID under which the channel initiator is running. For each CA certificate in the key ring associated with the entity, define a CNF with an IDNFILTER that exactly matches the subject DN of that CA certificate. This ensures that all certificates that the entity might use match at least one of these CNFs. This is because all such certificates must either be connected to the key ring associated with the entity, or must be issued by a CA for which a certificate is connected to the key ring associated with the entity.

Refer to the *z/OS Security Server RACF Security Administrator's Guide* for more information about the commands you use to manipulate CNFs.

z/OS *Defining a sender channel and transmission queue on QMA on z/OS*

Use the **DEFINE CHANNEL** and **DEFINE QLOCAL** commands to set up the required objects.

Procedure

On QMA, issue commands like the following example:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QMB.MACH.COM) XMITQ(QMB)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Sender channel using TLS from QMA to QMB')

DEFINE QLOCAL(QMB) USAGE(XMITQ)
```

Results

A sender channel, TO.QMB, and a transmission queue, QMB, are created.

z/OS *Defining a receiver channel on QMB on z/OS*

Use the **DEFINE CHANNEL** command to set up the required object.

Procedure

On QMB, issue a command like the following example:

```
DEFINE CHANNEL(TO.QMB) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS to QMB')
```

Results

A receiver channel, TO.QMB, is created.

z/OS *Starting the sender channel on QMA on z/OS*

If necessary, start a listener program and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: If you have not already done so, start a listener program on QMB.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener.
2. Optional: If any SSL/TLS channels have run previously, issue the command `REFRESH SECURITY TYPE(SSL)`.

   This ensures that all the changes made to the key repository are available.
3. Start the channel on QMA, using the command `START CHANNEL(TO.QMB)`.

Results

The sender channel is started.

z/OS *Exchanging self-signed certificates on z/OS*

Exchange the certificates you previously extracted. If you use FTP, use the correct format.

Procedure

Transfer the CA part of the QM1 certificate to the QM2 system and vice versa, for example, by FTP.

If you transfer the certificates using FTP, you must do so in the correct format.

Transfer the following certificate types in *binary* format:

- DER encoded binary X.509
- PKCS #7 (CA certificates)
- PKCS #12 (personal certificates)

Transfer the following certificate types in ASCII format:

- PEM (privacy-enhanced mail)
- Base64 encoded X.509

z/OS *Defining a sender channel and transmission queue on QM1 on z/OS*

Use the **DEFINE CHANNEL** and **DEFINE QLOCAL** commands to set up the required objects.

Procedure

On QM1, issue commands like the following example:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')

DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

The CipherSpecs at each end of the channel must be the same.

Only the SSLCIPH parameter is mandatory if you want your channel to use TLS. See “CipherSpecs e CipherSuites no IBM MQ” on page 43 for information about the permitted values for the SSLCIPH parameter.

Results

A sender channel, QM1.TO.QM2, and a transmission queue, QM2, are created.

z/OS *Defining a receiver channel on QM2 on z/OS*

Use the **DEFINE CHANNEL** command to set up the required object.

Procedure

On QM2, issue a command like the following example:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from QM1 to QM2')
```

The channel must have the same name as the sender channel you defined in “Defining a sender channel and transmission queue on QM1 on z/OS” on page 322, and use the same CipherSpec.

z/OS *Starting the sender channel on QM1 on z/OS*

If necessary, start a listener program and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: If you have not already done so, start a listener program on QM2.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener
2. Optional: If any SSL/TLS channels have run previously, issue the command REFRESH SECURITY TYPE(SSL).

   This ensures that all the changes made to the key repository are available.
3. On QM1, start the channel, using the command `START CHANNEL(QM1.TO.QM2)`.

Results

The sender channel is started.

z/OS *Refreshing the SSL or TLS environment on z/OS*

Refresh the TLS environment on queue manager QMA using the **REFRESH SECURITY** command.

Procedure

On QMA, enter the following command:

```
REFRESH SECURITY TYPE(SSL)
```

This ensures that all the changes made to the key repository are available.

z/OS *Allowing anonymous connections on a receiver channel on z/OS*

Use the **ALTER CHANNEL** command to make SSL or TLS client authentication optional.

Procedure

On QMB, enter the following command:

```
ALTER CHANNEL(TO.QMB) CHLTYPE(RCVR) SSLCAUTH(OPTIONAL)
```

z/OS *Starting the sender channel on QM1 on z/OS*

If necessary, start the channel initiator, start a listener program, and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: if you have not already done so, start the channel initiator.
2. Optional: If you have not already done so, start a listener program on QM2.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener
3. Optional: If the channel initiator was already running or any SSL/TLS channels have run previously, issue the command REFRESH SECURITY TYPE(SSL).

   This ensures that all the changes made to the key repository are available.
4. On QM1, start the channel, using the command `START CHANNEL(QM1.TO.QM2)`.

Results

The sender channel is started.

z/OS *Starting the sender channel on QMA on z/OS*

If necessary, start the channel initiator, start a listener program, and refresh security. Then start the channel using the **START CHANNEL** command.

Procedure

1. Optional: If you have not already done so, start the channel initiator.
2. Optional: If you have not already done so, start a listener program on QMB.

   The listener program listens for incoming network requests and starts the receiver channel when it is needed. For information about how to start a listener, see Starting a channel listener.
3. Optional: If the channel initiator was already running or if any SSL/TLS channels have run previously, issue the command `REFRESH SECURITY TYPE(SSL)`.

   This ensures that all the changes made to the key repository are available.
4. Start the channel on QMA, using the command `START CHANNEL(TO.QMB)`.

Results

The sender channel is started.

z/OS *Modifying elliptic curve key length on z/OS*

How you modify the GSK_CLIENT_ECURVE_LIST environment variable, to set the list of elliptic curves or supported groups that are specified by the client, as a string consisting of one or more 4-character values in order of preference for use.

Important: You must apply the fix in z/OS APAR OA61783 to permit certain elliptic curves to be made effective by the operating system, when using TLS 1.0, TLS 1.1 and/or TLS 1.2 negotiated connections.

You can set this TLS environment variable in the channel initiator startup JCL, using the CEEOPTS DD statement:

```
CEEOPTS DD DSN=<dataset-name>,DISP=SHR
```

In the dataset referenced above, specify the list that you want to use, for example:

```
ENVAR("GSK_CLIENT_ECURVE_LIST=002300240025")
```

Important: Do not use this CEEOPTS statement with in-stream data, as this prevents the environment variable from being set for all TLS tasks using that statement.

Ensure you reference a sequential dataset, or partitioned dataset member, to allow this to work when using an SSLTASKS value greater than one.

You can also use the server analogue equivalent of GSK_CLIENT_ECURVE_LIST, which is GSK_SERVER_ALLOWED_KEX_ECURVES. See Limiting key exchange elliptic curves for more information.

In addition, see Table 5 in Cipher suite definitions for a list of valid 4-character elliptic curve and supported groups specifications.

The default specification is `00210023002400250019`. If TLS V1.3 is enabled, `0029 (x25519)` is appended to the end of the default list.

Identificando e autenticando usuários

É possível identificar e autenticar usuários usando certificados X.509 , a estrutura MQCSP ou em vários tipos de programas de saída de usuário

Usando certificados X.509

É possível identificar e autenticar usuários usando certificados X.509 com o parâmetro **SET CHLAUTH** command e **SSLPEER** . O parâmetro **SSLPEER** especifica um filtro a ser usado para comparar com o Nome distinto do assunto do certificado do gerenciador de filas ou cliente peer na outra extremidade do canal.

Para obter mais informações sobre como usar o comando **SET CHLAUTH** e o parâmetro **SSLPEER** , consulte SET CHLAUTH

Os certificados digitais podem ser revogados pelas Autoridades de certificação. É possível verificar o status de revogação de certificados que usam OCSP ou CRLs nos servidores LDAP, dependendo da plataforma. Para obter informações adicionais, consulte “Trabalhando com Certificados Revogados” na página 345.

Usando a estrutura MQCSP

A estrutura de parâmetros de segurança de conexão MQCSP é especificada em uma chamada MQCONNX. Essa estrutura pode conter credenciais fornecidas pelo aplicativo. O aplicativo pode fornecer um ID do usuário e senha na estrutura MQCSP. No IBM MQ 9.3.4, os aplicativos também podem fornecer um token de autenticação. Se necessário, o MQCSP pode ser alterado em uma saída de segurança..

Aviso: As credenciais em uma estrutura MQCSP às vezes são enviadas pela rede em texto simples. Para assegurar que as credenciais do aplicativo cliente sejam protegidas, consulte “Proteção de senha do MQCSP” na página 32

Para obter mais informações, consulte o “Identificando e autenticando usuários usando a estrutura MQCSP” na página 327 e o “Trabalhando com tokens de autenticação.” na página 330.

Linux AIX Em AIX e Linux, o ID do usuário e a senha especificados na estrutura MQCSP podem ser autenticados usando o sistema operacional ou o Pluggable Authentication Method (PAM). O PAM fornece um mecanismo geral para a autenticação do usuário que oculta os detalhes dos serviços Para obter informações adicionais, consulte “Usando o Pluggable Authentication Method (PAM)” na página 357.

Implementando a identificação e a autenticação em saídas

É possível identificar e autenticar usuários usando vários tipos de programa de saída de usuário Para obter informações adicionais, consulte “Implementando identificação e autenticação em saídas de segurança” na página 328, “Mapeamento de identidade em saídas de mensagem” na página 329 e “Mapeamento de identidade na saída de API e saída cruzada da API” na página 329.

Usuários Privilegiados

Um usuário privilegiado é aquele que tem total autoridade administrativa para o IBM MQ.

Além dos usuários listados na tabela a seguir, há alguns objetos e autorizações para os quais muito cuidado deve ser tomado ao conceder acesso, para assegurar a integridade e a segurança do gerenciador de filas. Um exame detalhado extra deve ser aplicado ao conceder qualquer uma das autorizações a seguir:

- Quaisquer autorizações para objetos SYSTEM
- Autorizações de administração para criar, alterar e excluir objetos.

  z/OS No z/OS, essa autorização é a segurança de comando e a autoridade de segurança do recurso de comando para emitir comandos DEFINE, ALTER e DELETE.

  Multi Em todas as outras plataformas, essas autorizações são autorizações de administração, como `+crt`, `+chg` e `+dlt`.
- Autorização de administração para limpar filas.

z/OS No z/OS, essa autorização é a segurança de comando e a autoridade de segurança do recurso de comando para emitir comandos CLEAR.

Multi Em todas as outras plataformas, essa autorização é `+clr`.

- Autorizações de administração para parar canais, restaurar ou confirmar mensagens.

  z/OS No z/OS, essa autorização é a segurança de comando e a autoridade de segurança do recurso de comando para emitir comandos, como RESET CHANNEL, START CHANNEL e STOP CHANNEL.

  Multi Em todas as outras plataformas, essas autorizações são `+ctrl` e `+ctrlx`.
- Autorização do MQI de usuário alternativo que permite que os aplicativos escalem privilégios para verificações de autorização.

  z/OS No z/OS, essa autorização é qualquer autoridade concedida aos perfis de segurança de usuário alternativo.

  Multi Em todas as outras plataformas, essa autorização é `+altusr`.
- Autorizações de contexto que permitem que os aplicativos mudem o contexto de segurança de mensagens.

  z/OS No z/OS, essa autorização é qualquer autoridade concedida aos perfis de segurança de contexto.

  Multi Em todas as outras plataformas, essas autorizações são `+setall` e `+setid`.

Como um principal geral, os aplicativos de sistema de mensagens devem ter concedidos somente autorizações básicas de MQI para as filas ou os tópicos que são necessários. Os canais de MCA que são executados sob um MCAUSER não privilegiado e certos outros tipos especiais de aplicativos, como manipuladores de filas de mensagens não entregues, podem requerer autorizações adicionais não normalmente concedidas a aplicativos para operar corretamente.

Tabela 67. Usuários privilegiados por plataforma

Plataforma	**Usuários Privilegiados**
Sistemas Windows	• SISTEMA • Membros do grupo mqm • Membros do grupo Administradores
Sistemas AIX and Linux	• Membros do grupo mqm
Sistemas IBM i	• Os perfis qmqm e qmqmadm • Todos os membros do grupo qmqmadm • Qualquer usuário definido com a configuração *ALLOBJ
z/OS	O ID do usuário sob o qual o inicializador de canais, o gerenciador de fila e os espaços de endereço de segurança de mensagem avançada estão em execução... Esses IDs de usuário não têm automaticamente autoridades administrativas integrais para o IBM MQ, mas são considerados privilegiados devido ao nível de acesso que é normalmente concedido a esses IDs de usuário.

Identificando e autenticando usuários usando a estrutura MQCSP

É possível especificar a estrutura de parâmetros de segurança de conexão do MQCSP em uma chamada MQCONNX. A estrutura MQCSP é a maneira primária para aplicativos que usam a interface da fila de mensagens (MQI) para controlar as credenciais que são usadas para autenticação.

A estrutura MQCSP contém credenciais que o serviço de autorização pode usar para identificar e autenticar o usuário..

A estrutura MQCSP pode ser modificada por saídas de segurança do cliente ou do lado do servidor, mesmo se o aplicativo não fornecer explicitamente a estrutura MQCSP. Um exemplo de aplicativo que não fornece explicitamente uma estrutura MQCSP é um aplicativo que usa IBM MQ classes for JMS.. Para obter um exemplo de uma saída de segurança do lado do cliente que insere um ID do usuário e senha na estrutura MQCSP, consulte “Saída de segurança do lado do cliente para inserir o ID do usuário e a senha (mqccred)” na página 84.

V 9.4.0 A estrutura MQCSP contém um ID do usuário e senha ou um token de autenticação. As restrições a seguir se aplicam às credenciais fornecidas na estrutura MQCSP:

- Um aplicativo ou saída deve fornecer um ID do usuário e uma senha ou um token de autenticação, mas não ambos.
- Apenas tokens de autenticação que atendem formatos e requisitos específicos podem ser usados para acessar o IBM MQ. Para obter mais informações sobre os requisitos para tokens de autenticação no IBM MQ, consulte “Requisitos para tokens de autenticação” na página 333..
- Se a identidade no token de autenticação deve ser adotada como o contexto para o aplicativo, o token deve fornecer uma solicitação do usuário adequada e o valor da solicitação deve ser um ID do usuário IBM MQ válido. Por exemplo, o nome do usuário deve estar em conformidade com o comprimento máximo e as restrições de caracteres especiais Para obter mais informações sobre como adotar um ID do usuário, consulte “Relacionamento entre as configurações MQCSP e AdoptCTX” na página 327.

Para obter mais informações sobre a estrutura MQCSP, consulte MQCSP-Parâmetros de segurança

Aviso: As credenciais em uma estrutura MQCSP para um aplicativo cliente são, às vezes, enviadas pela rede em texto simples. Para assegurar que as credenciais do aplicativo cliente sejam protegidas, consulte “Proteção de senha do MQCSP” na página 32

Relacionamento entre as configurações MQCSP e AdoptCTX

IBM MQ sempre autentica credenciais que são transmitidas na estrutura MQCSP se o recurso de autenticação de conexão estiver ativado. Após as credenciais serem autenticadas com sucesso, o IBM MQ pode adotar o ID do usuário para verificações de autorização subsequentes em operações executadas pelo aplicativo conectado. O ID do usuário nas credenciais do MQCSP será adotado se o objeto de informações sobre autenticação (AUTHINFO) referenciado pelo atributo **CONNAUTH** do gerenciador de filas for definido com **ADOPTCTX(YES)**.

O IBM MQ tem um limite no comprimento de IDs de usuário que ele pode usar para verificações de autorização Para obter mais informações sobre esses limites, consulte “IDs de Usuário” na página 93 Quando um ID do usuário transmitido na estrutura MQCSP é adotado, o IBM MQ se comporta de forma diferente, dependendo de outras opções de configuração:

- Ao usar a autenticação de conexão LDAP, o IBM MQ adota o ID do usuário que está no atributo de nome do usuário curto do registro LDAP do usuário. O atributo username abreviado é configurado usando o atributo **SHORTUSR** do objeto AUTHINFO.

  Por exemplo, se **SHORTUSR** for configurado como `'CN'` e o registro LDAP listar o usuário como `'CN=Test,SN=MQ,O=IBM,C=UK'`, o ID do usuário `Test` será usado.
- Ao usar a autenticação de conexão do S.O. ou a autenticação PAM, se ADOPTCTX for YES, o ID do usuário transmitido na estrutura MQCSP será truncado para atender o limite de ID do usuário de 12 caracteres de IBM MQ quando adotado como o contexto de conexão.

Se **ChlAuthEarlyAdopt** estiver ativado, o truncamento acontecerá após as credenciais do usuário terem sido autenticadas.

Se **ChlAuthEarlyAdopt** não estiver ativado, o truncamento acontecerá antes da adoção. No Windows, se o usuário for fornecido no formato `user@domain`, isso significará que o truncamento pode resultar em uma especificação de domínio que não é válida quando o usuário tiver menos de 12 caracteres.

Por exemplo, se um usuário `` `ibmmq@windowsdomain` `` for fornecido por meio do MQCSP, ele será truncado para `` `ibmmq@window` `` neste cenário.. Isso resulta no seguinte erro:

```
AMQ8074W: a autorização falhou porque o SID 'SID' não corresponde à entidade 'ibmmq@window'
```

Nessa base, se você passar um ID do usuário com mais de 12 caracteres, como um ID do usuário do domínio Windows no formato `user@domain`, por meio do MQCSP, deverá configurar **ChlAuthEarlyAdopt**=*Y* no arquivo qm.ini para evitar esse erro.

Como alternativa, use ADOPTCTX (NO) na configuração CONNAUTH AUTHINFO e use uma abordagem alternativa como uma regra CHLAUTH USERMAP, uma saída de segurança ou a configuração MCAUSER do objeto do canal para configurar o ID do usuário para o canal.

Implementando identificação e autenticação em saídas de segurança

É possível usar uma saída de segurança para implementar a autenticação unilateral ou mútua.

O principal objetivo da saída de segurança é permitir que o MCA de cada extremidade de um canal autentique o seu parceiro. Em cada extremidade de um canal de mensagens, e na extremidade do servidor de um canal MQI, um MCA geralmente age em lugar do gerenciador de filas ao qual está conectado. Na extremidade do cliente de um canal do MQI, um MCA geralmente age em nome do usuário do aplicativo do IBM MQ MQI client. Nesta situação, a autenticação mútua realmente ocorre entre dois gerenciadores de filas ou entre um gerenciador de filas e o usuário de um aplicativo do IBM MQ MQI client.

A saída de segurança fornecida (a saída de canal SSPI) ilustra como a autenticação mútua pode ser implementada trocando-se os tokens de autenticação que são gerados e, em seguida, verificados por um servidor de autenticação confiável, como o Kerberos. Para obter mais detalhes, consulte "O programa de saída do canal SSPI no Windows" na página 161.

A autenticação mútua pode ser implementa também pela tecnologia PKI (Public Key Infrastructure). Cada saída de segurança gera alguns dados aleatórios, assina-os utilizando a tecla privativa do gerenciador ou usuário de fila que está representando e envia os dados assinados a seu parceiro em uma mensagem de segurança. A saída de segurança do parceiro executa a autenticação verificando a assinatura digital com a tecla privativa do gerenciador ou usuário da fila. Antes de trocar as assinaturas digitais, as saídas de segurança podem ter que concordar o algoritmo para gerar uma compilação de mensagem, se existir mais de um algoritmo disponível para uso.

Quando uma saída de segurança envia os dados assinados a seu parceiro, também precisa enviar algum meio de identificar o gerenciador ou usuário da fila que está representando. Pode ser um Nome Distinto ou mesmo um certificado digital. Se for enviado um certificado digital, a saída de segurança do parceiro poderá validar o certificado, operando pela cadeia de certificados do certificado CA raiz. Isto garante a propriedade da tecla pública utilizada para verificar a assinatura digital.

A saída de segurança do parceiro poderá validar um certificado digital somente se tiver acesso a um repositório de chaves que contenha os certificados restantes na cadeia de certificados. Se um certificado digital do gerenciador ou usuário de fila não for enviado, deverá haver um disponível no repositório de chaves ao qual a saída de segurança do parceiro tenha acesso. A saída de segurança do parceiro não poderá verificar a assinatura digital a menos que encontre a tecla pública do assinante.

A Segurança da Camada de Transporte (TLS) usa técnicas PKI como as que acabaram de ser descritas. Para obter mais informações sobre como o SSL realiza a autenticação, consulte "Conceitos de TLS (Transport Layer Security)" na página 18.

Se o suporte do servidor de autenticação ou PKI não estiver disponível, poderão ser utilizadas outras técnicas. Uma técnica comum, que pode ser implementada em saídas de segurança, utiliza um algoritmo de chave simétrico.

Uma das saídas de segurança, saída A, gera um número aleatório e o envia em uma mensagem de segurança para sua saída de segurança parceira, a saída B. A saída B criptografa o número usando sua cópia de uma chave que é conhecida apenas pelas duas saídas de segurança. A saída B envia o número criptografado para a saída A em uma mensagem de segurança com um segundo número aleatório que a saída B gerou. A saída A verifica se o primeiro número aleatório foi criptografado corretamente, criptografa o segundo número aleatório utilizando sua cópia da chave e envia o número criptografado à saída B em uma mensagem de segurança. A saída B verifica então se o segundo número foi criptografado corretamente. Durante essa troca, se nenhuma saída de segurança estiver satisfeita com a autenticidade da outra, poderá instruir o MCA a fechar o canal.

Uma vantagem desta técnica é que nenhuma chave ou senha é enviada para a conexão de comunicações durante a troca. Uma desvantagem é que não fornece uma solução para o problema de como distribuir a chave compartilhada de forma segura. Uma solução para esse problema está descrita em "Implementando confidencialidade em programas de saída do usuário" na página 475. Uma técnica semelhante é utilizada no SNA para a autenticação mútua de duas LUs quando elas se ligam para formar uma sessão. A técnica está descrita no "Autenticação em nível de sessão" na página 128.

Todas as técnicas anteriores de autenticação mútua podem ser adaptadas para fornecer autenticação unilateral.

Mapeamento de identidade em saídas de mensagem

É possível usar saídas de mensagens para processar informações para autenticar um ID do usuário, mas pode ser melhor implementar a autenticação no nível do aplicativo.

Quando um aplicativo põe uma mensagem em uma fila, o campo *UserIdentifier* no descritor da mensagem contém um ID do usuário associado ao aplicativo. Porém, não há dados presentes para serem utilizados para autenticar o ID do usuário. Esses dados podem ser incluídos por uma saída de mensagem na extremidade de envio de um canal e verificados por uma saída de mensagem na extremidade receptora do canal. Os dados de autenticação podem ser uma senha criptografada ou uma assinatura digital, por exemplo.

Este serviço pode ser mais efetivo se implementado no nível do aplicativo. O requisito básico é que o usuário do aplicativo que recebe a mensagem seja capaz de identificar e autenticar o usuário do aplicativo que enviou a mensagem. É, portanto, natural considerar a implementação desse serviço no nível do aplicativo. Para obter mais informações, consulte "Mapeamento de identidade na saída de API e saída cruzada da API" na página 329.

Mapeamento de identidade na saída de API e saída cruzada da API

Um aplicativo que recebe uma mensagem deve ser capaz de identificar e autenticar o usuário do aplicativo que enviou a mensagem. Esse serviço é geralmente melhor implementado no nível do aplicativo. Saídas de API podem implementar o serviço de várias maneiras.

No nível de uma mensagem individual, identificação e autenticação é um serviço que envolve dois usuários, o emissor e o receptor da mensagem. O requisito básico é que o usuário do aplicativo que recebe a mensagem seja capaz de identificar e autenticar o usuário do aplicativo que enviou a mensagem. Observe que este requisito serve para autenticação de uma via, não de duas vias.

Dependendo de como seja implementado, os usuários e seus aplicativos podem precisar fazer interface, ou mesmo interagir, com o serviço. Além disso, quando e como o serviço será utilizado pode depender de onde os usuários e seus aplicativos estão localizados, e da natureza dos próprios aplicativos. É, portanto, natural considerar a implementação do serviço ao nível do aplicativo, ao invés de ao nível do link.

Se você considerar a implementação deste serviço ao nível do link, você pode precisar resolver questões tais como as seguintes:

- Em um canal de mensagens, como aplicar o serviço apenas às mensagens que precisam?
- Como habilitar usuários e seus aplicativos a fazer interface, ou interagir, com o serviço, se isso é um requisito?

- Em uma situação de multisalto, em que uma mensagem é enviada em mais de um canal de mensagens a caminho de seu destino, onde você chamará os componentes do serviço?

Aqui estão alguns exemplos de como o serviço de identificação e autenticação pode ser implementado no nível do aplicativo. O termo *saída API* significa tanto uma saída API, quanto uma saída cruzada da API.

- Quando um aplicativo coloca uma mensagem em uma fila, uma saída API pode adquirir um token de autenticação de um servidor de autenticação confiável, como Kerberos. A saída API pode incluir este token nos dados do aplicativo na mensagem. Quando a mensagem for recuperada pelo aplicativo receptor, uma segunda saída API pode pedir ao servidor de autenticação que autentique o emissor verificando o token.
- Quando um aplicativo põe uma mensagem em uma fila, uma saída API pode anexar os seguintes itens aos dados do aplicativo na mensagem:
  - O certificado digital do emissor
  - A assinatura digital do emissor

  Se diferentes algoritmos para gerar uma compilação da mensagem estiverem disponíveis para utilização, a saída API pode incluir o nome do algoritmo que ela utilizou.

  Quando a mensagem for recuperada pelo aplicativo receptor, uma segunda saída API pode executar as seguintes verificações:

  - A saída API pode validar o certificado digital verificando toda a cadeia de certificados até o certificado de CA raiz. Para fazer isto a saída da API deve ter acesso ao repositório de chaves que contém os certificados restantes na cadeia de certificados. Esta verificação proporciona garantia de que o emissor, identificado pelo Nome Distinto, seja o proprietário genuíno da chave pública contida no certificado.
  - A saída API pode verificar a assinatura digital utilizando a chave pública contida no certificado. Essa verificação autentica o emissor.

  O Nome Distinto do emissor pode ser enviado, ao invés do certificado digital inteiro. Neste caso, o repositório de chaves deve conter o certificado do emissor, de modo que a segunda saída API possa encontrar a chave pública do emissor. Outra possibilidade é enviar todos os certificados na cadeia de certificados.
- Quando um aplicativo põe uma mensagem em uma fila, o campo *UserIdentifier* no descritor da mensagem contém um ID do usuário associado ao aplicativo. O ID do usuário pode ser utilizado para identificar o emissor. Para ativar a autenticação, uma saída API pode anexar alguns dados, tal com uma senha criptografada, aos dados do aplicativo na mensagem. Quando a mensagem for recuperada pelo aplicativo receptor, uma segunda saída API pode autenticar o ID do usuário utilizando os dados que seguiram com a mensagem.

  Esta técnica pode ser considerada suficiente para mensagens que se originem em um ambiente controlado e confiável, e em circunstâncias em que um servidor de autenticação confiável ou suporte PKI não estejam disponíveis.

Linux V 9.4.0 AIX Trabalhando com tokens de autenticação.

No IBM MQ 9.4.0, os aplicativos clientes podem fornecer tokens para autenticar com um gerenciador de filas em execução no AIX ou Linux. O ID do usuário no token também pode ser usado para autorização para acessar os recursos do IBM MQ .

Os JWTs (JSON Web Tokens) adotam um modelo de identidade baseado em solicitações. A identidade e o controle de acesso são abstraídos em ideias de solicitações e emissores de tokens.

- Uma solicitação é um par de valores de nome que contém informações sobre um usuário e estabelece quem é o usuário, não o que ele pode fazer
- O emissor do token é um terceiro confiável ou um servidor que emite um token para um usuário com base somente na identidade do usuário O emissor do token não está preocupado com o que o usuário pode fazer

Um token é uma estrutura simples que contém reivindicações e pode ser facilmente transferido entre as partes pela Internet. Usar tokens para autenticação tem o benefício de gerenciamento de identidade centralizada. É possível usar um emissor de token confiável para que seus aplicativos possam se autenticar com muitos serviços sem registrar separadamente com cada serviço. Os tokens fornecem maior segurança, pois as credenciais não são enviadas para cada serviço, apenas para o emissor confiável.

Um JWT é definido por meio do padrão de Internet proposto RFC7519

Como os tokens funcionam com o IBM MQ

Tokens que são usados com IBM MQ devem ser JWTs válidos que foram assinados com um algoritmo que o IBM MQ suporta. O JWT deve ser assinado de acordo com o padrão do JSON Web Signature (JWS). Tokens que usam as tecnologias JSON Web Encryption (JWE) e JSON Web Key (JWK) JOSE não podem ser usados com IBM MQ. Para obter mais informações, consulte “Requisitos para tokens de autenticação” na página 333.

O aplicativo que fornece o token de autenticação pode executar em qualquer plataforma que suporte IBM MQ clients. O aplicativo deve ser gravado em C ou em Java, e conectar-se ao gerenciador de filas usando ligações do cliente Entretanto, o gerenciador de filas deve ser executado em AIX ou Linux.

O gerenciador de filas valida a assinatura do token com relação à chave pública do emissor confiável ou chave simétrica no repositório de chaves. Para configurar o gerenciador de filas, siga as etapas em “Configurando um gerenciador de filas para aceitar tokens de autenticação usando um terminal JWKS” na página 336 ou Configurando um gerenciador de filas para aceitar tokens de autenticação usando um keystore local.

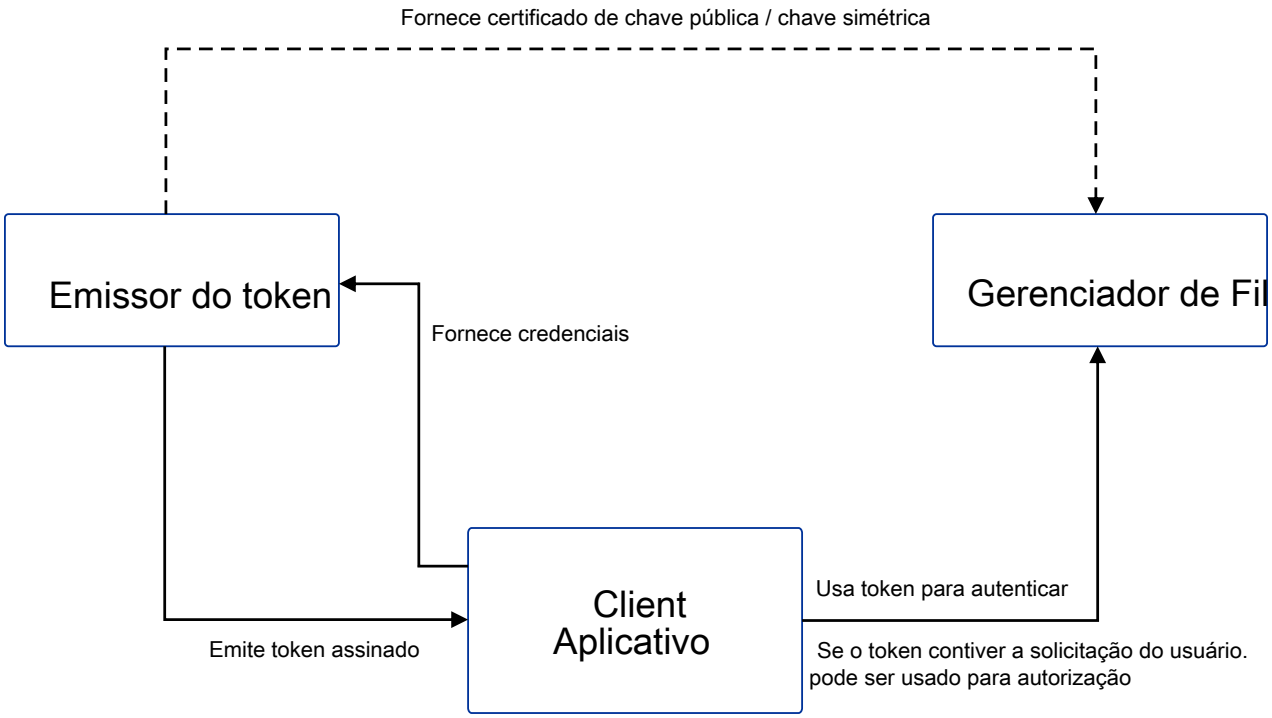
O emissor do token é a parte confiável que tem o acesso de segurança delegado, o que significa que eles verificam a identidade do usuário do aplicativo O gerenciador de filas verifica se um token de autenticação é válido e se o usuário autenticado está autorizado a acessar objetos do IBM MQ O gerenciador de filas pode, mas não precisa saber dos usuários antes de se conectarem primeiro com um token. O administrador do IBM MQ deve configurar autenticação e autorização para os aplicativos que se conectam ao gerenciador de filas e configurar os requisitos para o que os tokens devem conter.

O aplicativo cliente pode solicitar dinamicamente um token do emissor que ele usa para autenticação quando ele se conecta ao IBM MQ O aplicativo então usa a estrutura MQCSP, ou o equivalente na API escolhida, para passar o token para o gerenciador de filas quando ele se conecta.

Se o aplicativo não puder ser mudado para solicitar um token de autenticação e apresentar o token para o gerenciador de filas quando ele se conectar, uma saída de segurança poderá ser usada como alternativa para fornecer um token na estrutura MQCSP.

Se o token atender aos requisitos para tokens de autenticação e a assinatura do token for válida, a conexão será estabelecida.. O gerenciador de filas também pode usar o ID do usuário contido no token para verificações de autorização para acessar recursos do IBM MQ se a solicitação do usuário opcional estiver contida no token. A solicitação do usuário é a solicitação dentro do token que contém o ID do usuário que o gerenciador de filas adota para verificações de autorização. Esse nome da solicitação de usuário é especificado com o atributo **UserClaim** na sub-rotina **AuthToken** do arquivo `qm.ini` ..

Para obter mais informações, consulte “Usando tokens de autenticação em um aplicativo” na página 341 e MQCSP-Parâmetros de segurança



O diagrama mostra um exemplo básico do fluxo esperado para usar tokens com IBM MQ. O ciclo de vida esperado é o seguinte:

- O token é emitido para um aplicativo pelo emissor confiável.. Para obter mais informações, consulte Requisitos para tokens de autenticação
- O aplicativo passa o token para o gerenciador de fila ao se conectar Para obter mais informações, consulte Usando tokens de autenticação em um aplicativo
- O gerenciador de filas valida a assinatura do token com relação à chave pública do emissor confiável ou chave simétrica no repositório de chaves. Para configurar o gerenciador de filas, siga as etapas em "Configurando um gerenciador de filas para aceitar tokens de autenticação usando um terminal JWKS" na página 336
- Se o token de autenticação contiver uma solicitação do usuário válida, o usuário no token poderá ser adotado para verificações de autorização para acessar os recursos do IBM MQ Para obter mais informações, consulte Adotando usuários para autorização..
- O administrador do IBM MQ gerencia os certificados do emissor do token confiável Quando o certificado expirar, um novo certificado deverá ser obtido do emissor do token e incluído no repositório de chaves.
- Se você configurou seu gerenciador de fila e o aplicativo estiver se conectando, mas encontrar problemas com o token, consulte Resolução de problemas do token de autenticação e Códigos de erro de autenticação do token.

O IBM MQ funciona com qualquer emissor de token que fornece tokens que estão em conformidade com os padrões JWT e JWS

Se você ainda não estiver usando tokens, mas desejar entender o que está envolvido na criação de um servidor de token, consulte o Guia de Introdução para o projeto Keycloak de software livre.

Referências relacionadas

Sub-rotina AuthToken do arquivo `qm.ini`

Linux V 9.4.0 AIX

Requisitos para tokens de autenticação

Requisitos de validação, estrutura e algoritmos para tokens de autenticação usados com o IBM MQ

Requisitos

Os tokens de autenticação que são usados com o IBM MQ devem atender aos requisitos a seguir:

- O comprimento do token não deve exceder o comprimento máximo de 8192 caracteres. Para obter mais informações, consulte TokenLength (MQLONG) para MQCSP
- A estrutura e a codificação do token são válidas conforme definidas pela especificação JSON Web Token (JWT) em RFC7519e a especificação JSON Web Signature (JWS) em RFC7515.
- Os parâmetros de cabeçalho do token necessários especificados em Tabela 68 na página 334 estão presentes e os valores dos parâmetros são válidos.
- As solicitações de carga útil necessárias especificadas em Tabela 69 na página 335 estão presentes e os valores das solicitações são válidos..
- O token é assinado com um algoritmo no Tabela 70 na página 335 que IBM MQ suporta.
- O valor da solicitação de expiração (**exp**) é posterior ao horário atual
- Se a solicitação não anterior (**nbf**) estiver presente, o valor será anterior ao horário atual.
- Se uma solicitação do usuário estiver presente, o valor deverá atender aos requisitos para "IDs do usuário em tokens de autenticação.." na página 336..

Estrutura do token

O IBM MQ aceita JWTs que estão em conformidade com o padrão RFC7519 O JWT deve ser assinado e codificado de acordo com o padrão JWS definido em RFC7515.

O IBM MQ espera que o token JWS seguro contenha os três componentes a seguir:

Cabeçalho JOSE

Um objeto JSON que contém parâmetros que descrevem o tipo de token e os algoritmos criptográficos usados para proteger seu conteúdo.

O exemplo de cabeçalho a seguir declara que o objeto codificado é um JWT e que o cabeçalho e a carga útil são protegidos usando o algoritmo HMAC SHA-256 .

```
{
 "typ":"JWT",
 "alg":"HS256"
}
```

Carga útil do JWS

Um objeto JSON que contém solicitações, conforme especificado no padrão JWT Cada membro do objeto JSON é uma solicitação.. As solicitações podem declarar a identidade do emissor do token, ou o ID do usuário do portador.

```
{
 "exp": 1685529153,
 "nbf": 1685528150,
 "AppUser": "MyUserName"
 }
```

Assinatura JWS

Utilizado para validar que o token é emitido por um emissor confiável

Esses componentes são representados no token assegurado JWS como sequências base64url-encoded separadas por um ponto ('.').

Um token de autenticação que está em conformidade com o padrão JWS é assinado para permitir que a autenticidade do token seja validada, mas ele não é criptografado Portanto, ele pode ser lido e possivelmente reutilizado por qualquer pessoa que tenha acesso ao token. Configure a conexão com o gerenciador de filas para assegurar que a autenticação seja protegida usando a criptografia quando ela for enviada pela rede, por exemplo, usando TLS. Para obter mais informações sobre as opções para proteger credenciais que são fornecidas por um aplicativo, consulte Proteção de senha MQCSP.

O IBM MQ suporta os seguintes parâmetros e solicitações no cabeçalho e a carga útil de tokens de autenticação. Todos os parâmetros ou solicitações adicionais em um token são ignoradas Se um token contiver mais de um parâmetro ou solicitação com o mesmo nome, o último parâmetro ou solicitação com o nome duplicado será usado.

Tabela 68. Descrições do parâmetro do cabeçalho do token..

Parte do token	Nome do Parâmetro	Tipo de Dados	Necessário	Descrição
Cabeçalho	**typ**	Sequência	Sim	O tipo de token. O valor desse parâmetro deve ser "JWT".
	alg	Sequência	Sim	O algoritmo usado para proteger o cabeçalho e a carga útil O valor desse parâmetro deve ser um dos algoritmos em Tabela 70 na página 335.

Tabela 69. Descrições de solicitações de carga útil do token				
Parte do token	**Nome do Parâmetro**	**Tipo de Dados**	**Necessário**	**Descrição**
Carga Útil	**exp**	Integer	Sim	O tempo de expiração do token expresso como o número de segundos desde 1 de janeiro de 1979, 00:00 Hora Universal Coordenada. O token não é aceito após esse tempo
	nbf	Integer	No	O tempo, expresso como o número de segundos desde 1 de janeiro de 1979, 00:00 Hora Universal Coordenada antes do qual o token não é aceito.
	O nome da solicitação do usuário especificou no campo **UserClaim** da sub-rotina **AuthToken** no arquivo `qm.ini` ..	Sequência	Necessário apenas se a solicitação do usuário no token for usada para autorização	O nome da solicitação que contém o ID do usuário que é adotado para verificações de autorização. Por exemplo, se seu token tiver a solicitação do usuário `"AppUser": "MyUserName"`, você deverá especificar **UserClaim**=`AppUser` na sub-rotina **AuthToken** do arquivo `qm.ini` .

Para obter um bom exemplo de um token codificado e decodificado, consulte a página depurador no website `jwt.io` .

Algoritmos

O IBM MQ suporta um subconjunto de algoritmos que estão incluídos na especificação JSON Web Algorithms (JWA) para tokens seguros JWS .

Tabela 70. Algoritmos da Web JSON (JWA) suportados pelo IBM MQ para tokens protegidos JWS	
alg valor do parâmetro	**Algoritmo MAC ou Assinatura Digital**
HS256	HMAC usando SHA-256
HS384	HMAC usando SHA-384
HS512	HMAC usando SHA-512
RS256	RSASSA-PKCS1-v1_5 usando SHA-256
RS384	RSASSA-PKCS1-v1_5 usando SHA-384
RS512	RSASSA-PKCS1-v1_5 usando SHA-512

Requisitos de certificado de chave assimétrica

Se um token for assinado com uma chave assimétrica, o certificado de chave pública do emissor de token deverá estar no repositório de chaves que o gerenciador de filas usa para autenticação de token... Quando o token de autenticação é recebido, o certificado deve estar dentro de seu período de validade Nenhuma verificação é feita para assegurar que o certificado do emissor do token não foi revogado.

IDs do usuário em tokens de autenticação..

Se o gerenciador de filas estiver configurado para adotar o ID do usuário que está contido na solicitação do usuário de um token de autenticação como o contexto para o aplicativo, o ID do usuário que é adotado deverá atender aos seguintes requisitos:

- Pode conter até 12 caracteres.
- Ele deve iniciar com um dos seguintes caracteres:

  A-Z a-z
- Ele pode conter qualquer um dos seguintes caracteres:

  0-9 A-Z a-z +,-. : = _
- Ele não deve ser um dos IDs de usuário reservados UNKNOWN e NOBODY

Tarefas relacionadas

Configurando um gerenciador de filas para aceitar **AuthTokens**

Referências relacionadas

Sub-rotina AuthToken do arquivo `qm.ini`

Linux V 9.4.0 AIX Configurando um gerenciador de filas para aceitar tokens de autenticação usando um terminal JWKS

Configure seu gerenciador de filas do IBM MQ em execução no AIX ou no Linux para autenticar usuários e aplicativos com tokens de autenticação usando um terminal JWKS

Antes de começar

Para obter mais informações sobre como os tokens funcionam com o IBM MQ, consulte Trabalhando com tokens de autenticação.

Antes de configurar seu gerenciador de filas, verifique se o objeto AUTHINFO que é referido no atributo **CONNAUTH** do gerenciador de filas é do tipo `IDPWOS`. A autenticação do token está disponível apenas quando o gerenciador de filas está configurado para verificação de ID do usuário e senha do S.O.

Verifique se o atributo **SecurityPolicy** da sub-rotina de Serviço não está configurado como `Group` A autenticação do token não estará disponível se o **SecurityPolicy** estiver explicitamente configurado como `Group` Se**SecurityPolicy** está configurado para `Grupo` , remova o**SecurityPolicy** atributo da sub-rotina Service e, em seguida, reinicie o gerenciador de filas.

Sobre esta tarefa

Os aplicativos podem se autenticar com o gerenciador de filas usando tokens IBM MQ aceita JSON Web Tokens (*JWTs*) de emissores confiáveis que seguem o padrão de Internet proposto RFC7519. É possível usar tokens para autenticar uma identidade, que então pode ser adotada para futuras verificações de autorização.

A maneira mais simples de configurar seu gerenciador de filas para aceitar tokens é apontar para um terminal JWKS, conforme descrito abaixo. Se o seu serviço de autenticação não fornecer esse terminal ou o JWKS for inadequado por outros motivos, consulte “Configurando um gerenciador de filas para aceitar tokens de autenticação usando um keystore local” na página 337.

Procedimento

1. Peça ao administrador do servidor de autenticação estes detalhes:
   - O terminal JWKS correto (URL).
   - Qual certificado este servidor usa para criptografar o tráfego HTTP e / ou qual autoridade assina este certificado.

Importante: Você deve sempre fornecer informações do JWKS sobre TLS/HTTPS e precisa dessas informações para assegurar que o gerenciador de filas possa confiar na conexão.

2. Configure o gerenciador de filas para criar conexões https de saída fornecendo um **HTTPSKeyStore** no arquivo qm.ini .

   Para obter mais informações, consulte

   - A explicação do HTTPSKeyStore no arquivo `qm.ini`
   - “Criando um repositório de chaves para uso como um armazenamento confiável TLS” na página 344.

   Se o servidor de autenticação usar um certificado / CA sob medida, será necessário assegurar que ele esteja presente corretamente neste HTTPSKeyStore.

3. Configure o terminal JWKS definindo uma sub-rotina JWKS no arquivo de configuração qm.ini .

   A sub-rotina adicional fornece o seguinte:

   - **`issuername`**. Isso deve corresponder à solicitação '`iss`' que está presente em quaisquer tokens assinados por essa autoridade e geralmente é baseado na URL do serviço de autenticação.
   - **`endpoint`**. Este é o endereço a partir do qual o gerenciador de filas consulta chaves públicas usadas para validar assinaturas de token.
   - **`userclaim`**.Isso é opcional para identificar um campo customizado em tokens que deve ser usado para verificações de autoridade do IBM MQ quando um token tiver sido validado.

     Atenção: Isso deve estar presente se você pretende usar **ADOPTCTX**(YES) para tais conexões..

4. Quando as mudanças do arquivo `.ini` forem concluídas, emita o comando `REFRESH SECURITY TYPE(AUTHINFO)` ou reinicie o gerenciador de filas

   Se a configuração for bem-sucedida, os aplicativos poderão se conectar usando tokens assinados imediatamente.

   Se houver algum problema, por exemplo, não for possível entrar em contato com o serviço de autenticação para recuperar as chaves públicas, os problemas serão relatados no arquivo de log AMQERR01 para o gerenciador de fila.

Resultados

Um gerenciador de filas foi configurado com sucesso para aceitar tokens de autenticação usando um terminal JWKS.

Nota: As chaves são atualizadas periodicamente a partir do servidor de autenticação (a cada 15 minutos) e com mais frequência se um ID de chave desconhecido for apresentado por um aplicativo de conexão Geralmente, isso significa que nenhuma ação adicional de configuração do IBM MQ é necessária para atualizar certificados conforme eles expiram e são substituídos no lado do servidor. Para forçar uma atualização imediata emita o comando `REFRESH SECURITY TYPE(AUTHINFO)` a qualquer momento.

Conceitos relacionados

Resolução de problemas do token de autenticação

Tarefas relacionadas

Usando tokens de autenticação em um aplicativo

Referências relacionadas

Sub-rotina AuthToken do arquivo `qm.ini`

Linux V 9.4.0 AIX Configurando um gerenciador de filas para aceitar tokens de autenticação usando um keystore local

Configure o seu gerenciador de filas do IBM MQ para autenticar usuários e aplicativos com tokens de autenticação

Antes de começar

Quando possível, considere usar um terminal JWKS, consulte “Configurando um gerenciador de filas para aceitar tokens de autenticação usando um terminal JWKS” na página 336, em vez de configurar manualmente seus certificados de validação de token. O uso do JWKS geralmente simplifica a configuração inicial e a manutenção contínua.

Leia sobre como os tokens funcionam com IBM MQ em Trabalhando com tokens de autenticação.

Antes de configurar seu gerenciador de filas, verifique se o objeto AUTHINFO que é referido no atributo **CONNAUTH** do gerenciador de filas é do tipo IDPWOS. A autenticação do token está disponível apenas quando o gerenciador de filas está configurado para verificação de ID do usuário e senha do S.O.

Verifique se o atributo **SecurityPolicy** da sub-rotina de Serviço não está configurado como Group A autenticação do token não estará disponível se o **SecurityPolicy** estiver explicitamente configurado como Group Se **SecurityPolicy** estiver configurado como Group, remova o atributo **SecurityPolicy** da sub-rotina Service, em seguida, reinicie o gerenciador de filas.

Sobre esta tarefa

Em IBM MQ 9.3.4 , os aplicativos podem se autenticar com o gerenciador de filas usando tokens. IBM MQ aceita JSON Web Tokens (*JWTs*) de emissores confiáveis que seguem o padrão de Internet proposto RFC7519. É possível usar tokens para autenticar uma identidade, que então pode ser adotada para futuras verificações de autorização.

Configure seu gerenciador de filas para aceitar tokens salvando o certificado de chave pública ou a chave simétrica do emissor confiável no repositório de chaves do gerenciador de filas. Inclua a sub-rotina AuthToken no arquivo qm.ini e atualize a configuração de segurança para que o gerenciador de fila seleciona a nova configuração

Você pode desejar configurar um keystore local em vez de usar o JWKS em um ambiente de teste ou quando a conectividade direta com seu servidor de autenticação a partir de seu gerenciador de filas não for possível Também é possível definir um keystore local, além de quaisquer terminais JWKS

Nota: Quando um terminal JWKS e um keystore local fornecem um emissor e um KID correspondentes para um token apresentado, a chave fornecida pelo terminal JWKS é usada preferencialmente.

Nessas situações, configure o keystore local conforme a seguir:

Procedimento

1. Crie o repositório de chaves..
   a) Crie um repositório de chaves para o certificado de chave pública ou a chave simétrica recebida do emissor confiável. É possível usar um repositório de chave CMS com a extensão do arquivo .kdb ou um repositório de chaves PKCS#12 com a extensão do arquivo .p12.

   Emita o comando a seguir para criar um repositório de chaves CMS :

   ```
   runmqakm -keydb -create -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
   -type cms
   ```

   Se o comando **runmqakm** retornar um erro, consulte runmqakm -keydb. Se o comando for concluído com êxito, use o comando ls para listar o conteúdo do diretório::

   ```
   ls -l /var/mqm/qmgrs/qm1/tokenissuer
   ```

   Os seguintes arquivos são exibidos:

   ```
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.crl
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.kdb
   -rw------- 1 adminuser mqm 88 Feb 22 07:50 key.rdb
   ```

b) Se necessário, altere a propriedade do grupo para os arquivos de repositório de chaves criados para que o grupo mqm possa receber acesso de leitura. Inicialmente, apenas o usuário administrador que executou o comando tem acesso aos arquivos criados

```
chgrp mqm /var/mqm/qmgrs/qm1/tokenissuer/key.*
```

c) Mude o modo dos arquivos do repositório de chaves para incluir permissões de leitura para o grupo mqm. Por exemplo, o seguinte comando inclui permissões de leitura / gravação para o proprietário do arquivo e permissão somente leitura para o grupo.

```
chmod 640 /var/mqm/qmgrs/qm1/tokenissuer/key.*
```

2. Criptografe a senha do repositório de chaves com o comando **runqmcred** e salve a sequência criptografada em um arquivo

a) Crie um arquivo para conter a chave inicial usada para criptografar a senha do repositório de chaves.

O arquivo deve conter a chave inicial como uma única linha de texto O comprimento máximo da chave inicial é de 256 bytes Se você já tiver configurado uma chave inicial para o gerenciador de filas usando o atributo do gerenciador de filas **INITKEY** , copie o valor do atributo **INITKEY** no novo arquivo. Se você ainda não tiver configurado uma chave inicial para o gerenciador de fila, crie uma nova chave de criptografia exclusiva e inclua-a no arquivo de chaves inicial

Nota: Para obter mais informações, consulte INITKEY.. Se você não especificar a chave inicial, uma chave padrão será usada É mais seguro usar sua própria chave inicial.

Nota: Conceda as permissões mínimas necessárias no arquivo-chave inicial para manter o conteúdo do arquivo seguro O arquivo de chave inicial é usado apenas para criptografar a senha do repositório de chave Portanto, apenas os administradores que usam a chave inicial para criptografar senhas precisam de acesso ao arquivo de chave inicial de leitura de leitura.

b) Se a chave inicial do gerenciador de filas ainda não estiver configurada, configure o valor do atributo **INITKEY** do gerenciador de filas para a chave inicial criada na etapa “2.a” na página 339. Use o comando **ALTER QMGR** para configurar a chave inicial do gerenciador de filas Por exemplo:

```
ALTER QMGR INITKEY('myEncrypt10nK3y')
```

c) Emita o comando **runqmcred** para criptografar a senha do repositório de chaves.. Use o parâmetro **-sf** para especificar o caminho para o arquivo que contém a chave inicial

```
runqmcred -sf initial.key
```

Quando solicitado, insira a senha do repositório de chaves.. A senha criptografada é a saída pelo comando

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Enter password:
*************
<QM>!2!b5rb01sMzFzc1ClZeQMryruWFM3HSm8DKyEaZK7qzWY=!TrWdU57DCDXM0Qah99I/Lg==
```

Copie a sequência na última linha e salve-a em um arquivo

3. Use um dos métodos a seguir para incluir o certificado de chave pública ou chave simétrica do emissor do token no repositório de chaves.

- Para incluir o certificado de chave pública RSA no repositório de chaves, emita o comando a seguir:

```
runmqakm -cert -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile
```

- Para incluir uma chave simétrica codificada base64 no repositório de chaves, emita o comando a seguir:

```
runmqakm -secretkey -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile -format ascii
```

Em que *keylabel* é o rótulo a ser anexado ao certificado ou chave secreta e *keyfile* é o nome do arquivo que contém o certificado ou a chave secreta codificada base64 .

4. Inclua a sub-rotina **AuthToken** e os atributos a seguir no arquivo `qm.ini` :
   - O caminho para o repositório de chaves, especificado usando o atributo **KeyStore**
   - O arquivo que contém a senha para o repositório de chaves, especificado usando o atributo **KeyStorePwdFile**
   - O rótulo do certificado ou chave simétrica incluído na etapa “3” na página 339, especificado usando o atributo **CertLabel** .

   Por exemplo:

```
AuthToken:
   KeyStore=/var/mqm/qmgrs/qm1/tokenissuer/key.kdb
   KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
   CertLabel=rsakey
```

   Em que `key.kdb` é o nome do repositório de chaves criado na etapa “1.a” na página 338e `key.pw` é o arquivo que contém a senha criptografada para o repositório de chaves criado na etapa “2.c” na página 339.

   Para obter mais informações sobre a sub-rotina **AuthToken** , consulte a sub-rotina AuthToken do arquivo `qm.ini` .
5. Se o gerenciador de filas estiver configurado para adotar o ID do usuário que está contido na solicitação do usuário do token para uso em verificações de autorizações subsequentes, inclua o atributo **UserClaim** na sub-rotina **AuthToken** ..

   Para determinar se o gerenciador de filas está configurado para adotar o ID do usuário no token, emita o comando MQSC a seguir:

```
DISPLAY AUTHINFO(authinfo_name) ADOPTCTX
```

   Em que *authinfo_name* é o valor do atributo **CONNAUTH** do gerenciador de filas. Se o valor do atributo **ADOPTCTX** for YES, o gerenciador de filas será configurado para adotar o ID do usuário no token e o atributo **UserClaim** deverá ser especificado na sub-rotina **AuthToken**

   Configure o valor do atributo **UserClaim** para o nome da solicitação de token que contém o ID do usuário a ser adotado.. Por exemplo, se o token contiver a solicitação `"AppUser": "MyUserName"`, inclua a linha a seguir na sub-rotina **AuthToken** :

```
UserClaim=AppUser
```

6. Atualize a configuração de segurança do gerenciador de filas para que ele selecione a configuração do token do arquivo `qm.ini` . Emita o comando a seguir para iniciar o comando **runmqsc** :

```
runmqsc qm1
```

   em seguida, emita o comando MQSC a seguir:

```
REFRESH SECURITY TYPE(CONNAUTH)
```

Como proceder a seguir

Trabalhe com seus desenvolvedores para ajudá-los a entender como eles podem usar tokens em aplicativos para se autenticar com o gerenciador de fila

Conceitos relacionados

Resolução de problemas do token de autenticação

Tarefas relacionadas

Usando tokens de autenticação em um aplicativo

Referências relacionadas

Sub-rotina AuthToken do arquivo `qm.ini`

Linux V 9.4.0 AIX Obtendo um token de autenticação de seu emissor de token escolhido

Grave seu aplicativo para obter um token de autenticação de seu emissor de token escolhido quando ele se conectar a um gerenciador de filas do IBM MQ

Antes de começar

Consulte as informações em “Usando tokens de autenticação em um aplicativo” na página 341

Procedimento

- Como você obtém um token de autenticação e o conteúdo exato do token varia entre diferentes emissores de token.

  Grave seu aplicativo para interagir com seu emissor de token escolhido para solicitar e obter o token da autenticação O token de autenticação deve estar em conformidade com os requisitos IBM MQ para tokens de autenticação. Para obter mais informações sobre esses requisitos, consulte “Requisitos para tokens de autenticação” na página 333

  Se você pretende adotar um ID do usuário que está contido em uma solicitação de token como o contexto para o aplicativo, o token de autenticação também deve atender aos seguintes requisitos:

  – O token de autenticação deve conter uma solicitação que corresponda ao nome da solicitação do usuário na configuração de autenticação do token do gerenciador de filas

  – O valor da solicitação do usuário deve atender aos requisitos para os IDs do usuário em tokens de autenticação Para obter informações adicionais, consulte “IDs do usuário em tokens de autenticação..” na página 336.

Resultados

Agora você obteve um JWT formatado corretamente, que pode ser apresentado ao IBM MQ para validação

Tarefas relacionadas

Configurando um gerenciador de filas para aceitar **AuthTokens**

Referências relacionadas

Sub-rotina AuthToken do arquivo `qm.ini`

MQCSP-Parâmetros de segurança

Linux V 9.4.0 AIX Usando tokens de autenticação em um aplicativo

Grave seu aplicativo para fornecer um token de autenticação quando ele se conectar a um gerenciador de filas do IBM MQ

Antes de começar

No IBM MQ 9.4.0, os aplicativos podem fornecer um token de autenticação quando se conectam a um gerenciador de filas.

O pedido deve satisfazer os seguintes requisitos:

- Ele deve ser gravado em C ou Java (usando o IBM MQ classes for JMS/ Jakarta Messaging)

- Ele deve se conectar ao Gerenciador de Filas como um IBM MQ client Ou seja, o aplicativo deve se conectar ao gerenciador de filas em uma rede, em vez de usar ligações locais.
- Ele deve se conectar a um gerenciador de filas executado em AIX ou Linux.

Se o aplicativo não atender a esses requisitos, a conexão falhará e o código de razão MQRC_FUNCTION_NOT_SUPPORTED (2298) será retornado ao aplicativo.

O aplicativo que fornece o token de autenticação pode executar em qualquer plataforma que suporte IBM MQ MQI clients.

Os clientes que usam a reconexão automática do cliente não podem fornecer um token de autenticação quando eles se conectam Se um aplicativo fornecer um token de autenticação e especificar a opção MQCNO_RECONNECT ou MQCNO_RECONNECT_Q_MGR na estrutura MQCNO, a conexão falhará e o código de razão MQRC_RECONNECT_INCOMPATÍVEL (2547) será retornado ao aplicativo. Para obter mais informações sobre a reconexão automática do cliente, consulte Reconexão automática do cliente.

Se não for possível gravar o aplicativo para fornecer um token de autenticação devido a esses requisitos, será possível migrar seu aplicativo para usar tokens de autenticação usando uma saída de segurança do cliente. A saída de segurança do cliente pode ser gravada para configurar o token de autenticação na estrutura MQCSP.. Para obter mais informações sobre saídas de segurança, consulte Saídas de segurança em uma conexão do cliente.

No IBM MQ 9.4.0, os aplicativos clientes JMS podem fornecer diretamente um token ao se conectar (consulte “Obtendo um token de autenticação de seu emissor de token escolhido” na página 341). Antes do IBM MQ 9.4.0, os aplicativos Java podem fornecer um token indiretamente por meio de um programa de saída.. Para obter mais informações, consulte Classe Java MQCSP..

Sobre esta tarefa

Nota: Um token de autenticação que está em conformidade com o padrão JSON Web Signature (JWS) é assinado para permitir que a autenticidade do token seja validada, mas não é criptografada. Portanto, ele pode ser lido e possivelmente reutilizado por qualquer pessoa que tenha acesso ao token. Configure a conexão para o gerenciador de filas para assegurar que o token de autenticação seja protegido usando criptografia quando for enviado pela rede, por exemplo, usando TLS. Para obter mais informações sobre as opções para proteger credenciais fornecidas por um aplicativo, consulte “Proteção de senha do MQCSP” na página 32.

Antes de modificar os aplicativos para se conectarem usando um token, assegure:

- O gerenciador de filas foi configurado para aceitar tokens de autenticação seguindo as etapas em “Configurando um gerenciador de filas para aceitar tokens de autenticação usando um keystore local” na página 337
- Seu aplicativo pode obter um token válido conforme necessário de seu servidor de autenticação, consulte “Obtendo um token de autenticação de seu emissor de token escolhido” na página 341.

Para fornecer um token de autenticação quando o aplicativo se conectar a um gerenciador de fila do IBM MQ , inclua o processo a seguir:

Procedimento

- Para fornecer um token de autenticação de um aplicativo C (MQI):

  O aplicativo deve se conectar usando MQCONNX (em vez de MQCONN) e fornecer uma estrutura MQCSP :

  - O campo **AuthenticationType** deve ser configurado como MQCSP_AUTH_ID_TOKEN
  - A versão da estrutura deve ser configurada como MQCSP_VERSION_3
  - O campo **TokenPtr** ou **TokenOffset** deve referenciar seu token de autenticação
  - O campo **TokenLength** deve ser configurado para o comprimento do token de autenticação

  Código C de exemplo para se conectar a um gerenciador de filas usando MQCSP Versão 3 e token de autenticação:

```
MQCNO cno = {MQCNO_DEFAULT};   /* Connection options  */
MQCSP csp = {MQCSP_DEFAULT};   /* Security parameters */

char  token[MQ_CSP_TOKEN_LENGTH +1] = {0};  /* Authentication token string */

/* Set the connection options */
cno.SecurityParmsPtr = &csp;
cno.Version = MQCNO_VERSION_5;

/* Set the security parameters */
csp.Version = MQCSP_VERSION_3;
csp.AuthenticationType = MQCSP_AUTH_ID_TOKEN;
csp.TokenPtr = token;
csp.TokenLength = (MQLONG) strlen(token);

/* Connect to the queue manager */
MQCONNX(qmName,                 /* Queue manager name  */
        &cno,                   /* Connection options  */
        &hCon,                  /* Connection handle   */
        &compCode,              /* Completion code     */
        &reason);               /* Reason code         */
```

- Para fornecer um token de autenticação a partir de um aplicativo Java :

  Os aplicativos que usam o IBM MQ classes for JMS/Jakarta Messaging podem fornecer um token por meio de qualquer um dos métodos `createContext`ou `createConnection` , que usam um nome do usuário e uma senha.

  Para fornecer um token de autenticação, o:

  – **`UserID`** deve ser configurado como nulo ou uma cadeia vazia, ou seja, sem espaços, " "

  – O token é fornecido como a sequência **`Password`** ..

  Isso se aplica a todas as implementações de IBM MQ da interface `ConnectionFactory`

  Os formulários de parâmetro explícito, por exemplo, `createContext`(String **`userID`**, String **`password`**) podem ser usados ou as versões de parâmetro implícito, por exemplo, `createContext`().

  No último caso, o **`userID`** vazio e o Token **`Password`** devem ter sido fornecidos primeiro como propriedades no connection factory

  Exemplo de código Java para se conectar a um gerenciador de filas usando um token de autenticação:

```
// Obtain token from authentication provider here:

String myToken = "xxxxxxxxxxxxxxx";

// Acquire instance of an MQ connection Factory:

JmsFactoryFactory ff = JmsFactoryFactory.getInstance(WMQConstants.WMQ_PROVIDER);

JmsConnectionFactory cf = ff.createConnectionFactory();

// Configure any required CF properties here - e.g. MQ Channel details

// Connect to (and authenticate with) the queue manager:

context = cf.createContext(null, myToken); // NOTE - null userID indicates token being
provided
```

  Se a conexão falhar com o código de razão MQRC_NOT_AUTHORIZED (2035) ou MQRC_SECURITY_ERROR (2063), verifique o log de erro do gerenciador de filas para obter uma mensagem de erro contendo mais informações sobre a causa da falha. Para obter mais ajuda para diagnosticar problemas com tokens de autenticação, consulte Resolução de problemas de token de autenticação.

Resultados

O aplicativo agora está conectado ao Gerenciador de Filas Ele permanece conectado até ser desconectado, mesmo se o token que foi usado para autenticar expirar. Se o aplicativo se desconectar do

gerenciador de filas e precisar se reconectar, poderá ser necessário obter um novo token de autenticação com um tempo de expiração posterior antes que ele possa se reconectar

Tarefas relacionadas

Configurando um gerenciador de filas para aceitar **AuthTokens**

Referências relacionadas

Sub-rotina AuthToken do arquivo `qm.ini`

MQCSP-Parâmetros de segurança

Linux V 9.4.0 AIX Criando um repositório de chaves para uso como um armazenamento confiável TLS

Ao criar conexões TLS de saída, você deve criar um 'armazenamento confiável' simples que possa validar certificados assinados por um conjunto comum de autoridades de certificação (CAs). As conexões TLS de exemplo são um canal do cliente IBM MQ ou uma conexão HTTPS, conforme usado ao configurar alguns componentes do IBM MQ.

Sobre esta tarefa

Atenção: Decidir quais certificados e autoridades de certificação confiar em seu ambiente é uma etapa importante com implicações para a segurança da configuração de ponta a ponta. Este tópico é fornecido para ilustrar etapas comuns que permitem que os componentes do IBM MQ confiem no mesmo conjunto de certificados já configurados para seu sistema operacional; no entanto, em caso de dúvida, você deve discutir esse processo com seu administrador de segurança

A maioria dos sistemas operacionais baseados em UNIX e Linux possuem um local do sistema de arquivos contendo um conjunto 'confiável' de autoridades de certificação. Esse sistema de arquivos pode ter sido configurado com a instalação do sistema operacional, ou customizado pelo administrador do sistema (por exemplo, para incluir CAs internas pertencentes à organização). Os locais para esses arquivos variam, mas alguns valores comumente usados para sistemas operacionais populares são:

- AIX: `/var/ssl/cert.pem and/or /var/ssl/certs/*.crt`
- RHEL: `/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem`
- Ubuntu: `/etc/ssl/certs/*.pem`

Ao criar e configurar um keystore IBM MQ , é possível incluir facilmente todos os arquivos de certificado em um diretório, por exemplo, `/etc/ssl/certs`, em um banco de dados de chave do IBM MQ em um comando

Procedimento

1. Use o comando a seguir para incluir os arquivos de certificado do diretório `/etc/ssl/certs` :

```
runmqakm -cert -add -file /etc/ssl/certs/*.pem -db mykdb.p12 -stashed
```

2. Opcional: Em algumas situações, pode ser útil gerar um conjunto 'padrão' de certificados para o seu armazenamento confiável

   Os componentes de segurança do IBM MQ fornecidos com o produto fornecem um conjunto de certificados de CA 'padrão'.

   Nota: Esses certificados podem não ser atualizados com frequência ou ter tempos de vida relativamente curtos.

   Se desejar usar os certificados de autoridade de certificação pré-configurados de qualquer maneira, será possível gerar um armazenamento confiável usando os parâmetros **`populate`** e **`ibmcloudtrust`** no comando **`runmqakm`** :

```
runmqakm -keydb -create -db mqauto.p12 -genpw -stash -type pkcs12 -populate -ibmcloudtrust
```

Conceitos relacionados

Resolução de problemas do token de autenticação

Tarefas relacionadas

Usando tokens de autenticação em um aplicativo

Referências relacionadas

Sub-rotina AuthToken do arquivo `qm.ini`

Trabalhando com Certificados Revogados

Os certificados digitais podem ser revogados pelas Autoridades de certificação. É possível verificar o status de revogação de certificados que usam OCSP ou CRLs nos servidores LDAP, dependendo da plataforma.

Durante o handshake TLS, os parceiros de comunicação se autenticam com certificados digitais. A autenticação pode incluir a verificação de que o certificado recebido ainda pode ser confiável. As Autoridades de certificação (CAs) revogam certificados por muitas razões, incluindo:

- O proprietário mudou para uma organização diferente.
- A chave privada não é mais secreta

As CAs publicam os certificados pessoais revogados em uma CRL (Lista de Certificados Revogados). Os certificados de CA que foram revogados são publicados em uma ARL (Lista de Autoridades Revogadas).

Nas plataformas AIX, Linux, and Windows, o suporte SSL do IBM MQ verifica os certificados revogados usando o OCSP (Online Certificate Status Protocol) ou usando CRLs e ARLs em servidores LDAP (Lightweight Directory Access Protocol). OCSP é o método preferido.

IBM MQ classes for Java e IBM MQ classes for JMS não pode usar as informações do OCSP em um arquivo da tabela de definição de canal do cliente. No entanto, você pode configurar o OCSP conforme descrito em Usando Protocolo de Certificado Online.

No IBM i, IBM MQ o suporte SSL verifica os certificados revogados usando CRLs e ARLs somente em servidores LDAP

No z/OS, IBM MQ o suporte SSL verifica os certificados revogados usando CRLs e ARLs somente em servidores LDAP

Para mais informações sobre Autoridades de certificação, consulte “Certificados Digitais” na página 13.

Verificação de OCSP/CRL

A verificação do Online Certificate Status Protocol (OCSP)/Lista de Revogação de Certificado (CRL) é realizada com relação aos certificados recebidos remotos. O processo verifica toda a cadeia envolvida desde o certificado pessoal do sistema remoto até o certificado raiz.

Usando o openSSL para verificar a validação do OCSP

Se sua empresa usar openSSL para validar OCSP e, em seguida, você tentar usar uma conexão TLS do IBM Global Security Kit (GSKit) , receberá um aviso de status UNKNOWN.

Isso ocorre porque todos os certificados na cadeia, além da raiz, são verificados pelo GSKit para o status de revogação A operação GSKit está em conformidade com RFC 5280 e isso é descrito na política de confiança do GSKit . O algoritmo GSKit tenta todas as origens disponíveis para informações de revogação, conforme descrito no RFC 5280 e na Política de Confiança do GSKit

Como a verificação de OCSP/CRL funciona no IBM MQ?

O IBM MQ suporta dois mecanismos para controle de comportamento ao verificar certificados com relação a terminais OCSP ou CRL nomeados, seja na extensão do certificado ou conforme definido nos objetos AUTHINFO:

- Os atributos **OCSPCheckExtensions**, **CDPCheckExtensions** e **OCSPAuthentication** da sub-rotina de SSL do arquivo qm.ini e
- Usando o parâmetro SSLCRLNL do gerenciador de filas e as configurações AUTHINFO OCSP e CRLLDAP. Consulte ALTER AUTHINFO e ALTER QMGR para obter mais informações.

Atenção:

O comando ALTER AUTHINFO com **AUTHTYPE(OCSP)** não se aplica para uso em Gerenciadores de Filas IBM i ou z/OS. No entanto, pode ser especificado nas plataformas a serem copiadas na tabela de definição de canal do cliente (CCDT) para uso do cliente.

Os atributos de sub-rotina SSL **OCSPCheckExtensions** e **CDPCheckExtensions** controlam se o IBM MQ verificará um certificado com relação ao servidor OCSP ou CRL detalhado na extensão AIA do certificado.

Se não estiver ativado, o servidor OCSP ou CRL na extensão do certificado não será contatado.

Se os servidores OCSP ou CRL forem detalhados por meio de objetos AUTHINFO e referenciados usando o atributo SSLCRLNL **QMGR**, durante o processamento de revogação de certificado, o IBM MQ tentará entrar em contato com esses servidores.

Importante: Apenas um objeto OCSP AUTHINFO pode ser definido na lista de nomes SSLCRLNL.

Se:

OCSPCheckExtensions=NO e **CDPCheckExtensions**=*NO* estão configurados e
Nenhum servidor OCSP ou CRL é definido em objetos AUTHINFO

nenhuma verificação de revogação de certificados é executada.

Ao verificar o status de revogação de um certificado, o IBM MQ entra em contato com os servidores OCSP ou CRL nomeados na ordem a seguir, se ativados:

1. O servidor OCSP detalhado em um objeto **AUTHTYPE(OCSP)** e referenciado no atributo SSLCRLNL **QMGR**.
2. Os servidores OCSP detalhados na extensão AIA dos certificados, se **OCSPCheckExtensions**=*YES*.
3. Os servidores CRL detalhados na extensão **CRLDistributionPoints** dos certificados, se **CDPCheckExtensions**=*YES*.
4. Todos os servidores CRL detalhados em objetos **AUTHINFO(CRLLDAP)** e referenciados no atributo SSLCRLNL **QMGR**.

Ao verificar um certificado, se uma etapa resultar no servidor OCSP ou no servidor CRL retornando uma resposta definitiva REVOKED ou VALID para uma consulta do certificado, não serão realizadas verificações adicionais e o status do certificado será usado da maneira que ele foi apresentado para determinar se ele é confiável ou não.

Se um servidor OCSP ou servidor CRL retornar um resultado de UNKNOWN, o processamento continuará até que um servidor OCSP ou CRL retorne um resultado definitivo ou até que todas as opções sejam esgotadas.

O comportamento resultante de um certificado ser considerado revogado ou não, caso o status não possa ser determinado, é diferente para servidores OCSP e CRL:

- Para servidores CRL, se nenhum CRL puder ser obtido, o certificado será considerado NOT_REVOKED
- Para servidores OCSP, se nenhum status de revogação puder ser obtido de um servidor OCSP nomeado, o comportamento será controlado por meio do atributo **OCSPAuthentication** na sub-rotina SSL do arquivo qm.ini.

  É possível configurar esse atributo para bloquear uma conexão, permitir uma conexão ou permitir uma conexão com uma mensagem de aviso.

Será possível usar o atributo **SSLHTTPProxyName**=*string* na sub-rotina SSL dos arquivos qm.ini e mqclient.ini para as verificações de OCSP, se necessário. A sequência é o nome do host ou o endereço de rede do servidor Proxy HTTP que deve ser usado pelo GSKit para verificações do OCSP

É possível configurar o valor **OCSPTimeout** na sub-rotina SSL dos arquivos `qm.ini` ou `mqclient.ini` que configura o número de segundos para aguardar um respondente OCSP ao executar uma verificação de revogação.

ALW Certificados Revogados e OCSP

O IBM MQ determina qual respondente Online Certificate Status Protocol (OCSP) usar e manipula a resposta recebida. Pode ser necessário concluir etapas para tornar o respondente do OCSP acessível.

Nota: Estas informações se aplicam apenas ao IBM MQ nos sistemas AIX, Linux, and Windows.

Para verificar o status da revogação de um certificado digital usando OCSP, o IBM MQ pode usar dois métodos para determina com qual respondente OCSP entrar em contato:

- Usando a extensão de certificado AuthorityInfoAccess (AIA) no certificado a ser verificado.
- Usando uma URL especificada em um objeto de informação de autenticação ou especificada por um aplicativo cliente.

Uma URL especificada em um objeto de informações de autenticação ou por um aplicativo cliente que tem prioridade sobre uma URL em uma extensão de certificado de AIA.

Se a URL do respondente do OCSP estiver atrás de um firewall, reconfigure o firewall para que o respondente do OCSP possa ser acessado ou configure um servidor proxy do OCSP. Especifique o nome do servidor proxy usando a variável SSLHTTPProxyName na sub-rotina SSL. Nos sistemas do cliente, também é possível especificar o nome do servidor proxy usando a variável de ambiente MQSSLPROXY. Para obter mais detalhes, consulte as informações relacionadas.

Se você não estiver preocupado se os certificados TLS foram revogados, talvez porque esteja executando em um ambiente de teste, será possível configurar OCSPCheckExtensions para NO na sub-rotina SSL. Se você configurar essa variável, qualquer extensão de certificado AIA será ignorada. É possível que essa solução não seja aceita em um ambiente de produção, no qual você pode querer não permitir o acesso de usuários que possuírem certificados revogados.

A chamada para acessar o respondente do OCSP pode resultar em um dos três resultados a seguir:

válido
O certificado é válido.

Revogado
O certificado é revogado.

Desconhecido
Esse resultado pode surgir por um dos três motivos:

- O IBM MQ não pode acessar o respondente OCSP.
- O respondente OCSP enviou uma resposta, mas o IBM MQ não pode verificar a assinatura digital da resposta.
- O respondente do OCSP enviou uma resposta indicando que ele não possui nenhum dado de revogação para o certificado.

Se o IBM MQ receber um resultado do OCSP de `Desconhecido`, seu comportamento dependerá da configuração do atributo OCSPAuthentication. Para gerenciadores de filas, este atributo é mantido em um dos locais a seguir:

- Linux AIX Na sub-rotina SSL do arquivo `qm.ini` no AIX and Linux.
- Windows No registro do Windows.

Este atributo pode ser configurado usando o IBM MQ Explorer. Para clientes, o atributo é mantido na sub-rotina de SSL do arquivo de configuração do cliente.

Se um resultado `Desconhecido` for recebido e OCSPAuthentication estiver configurado como REQUIRED (o valor padrão), o IBM MQ rejeitará a conexão e emitirá uma mensagem de erro de tipo AMQ9716. Se as mensagens de evento do SSL do gerenciador de filas estiverem ativadas, uma

mensagem de evento SSL do tipo MQRC_CHANNEL_SSL_ERROR com ReasonQualifier configurado para MQRQ_SSL_HANDSHAKE_ERROR é gerada.

Se um resultado `Desconhecido` for recebido e OCSPAuthentication estiver configurado como OPTIONAL, o IBM MQ permitirá que o canal SSL seja iniciado e não são gerados avisos nem mensagens de eventos SSL.

Se um resultado `Desconhecido` for recebido e OCSPAuthentication estiver configurado como WARN, o canal SSL será iniciado, mas o IBM MQ emitirá uma mensagem de aviso do tipo AMQ9717 no log de erro. Se as mensagens de evento do SSL do gerenciador de filas estiverem ativadas, uma mensagem de evento SSL do tipo MQRC_CHANNEL_SSL_WARNING com ReasonQualifier configurado para MQRQ_SSL_UNKNOWN_REVOCATION é gerada.

Assinatura Digital das Respostas do OCSP

Um respondente do OCSP pode assinar suas respostas de uma de três maneiras. Seu respondente informará qual método é usado.

- A resposta do OCSP pode ser assinada digitalmente usando o mesmo certificado de CA que emitiu o certificado que estiver verificando. Nesse caso, não é necessário configurar nenhum certificado adicional; as etapas já concluídas para estabelecer a conectividade TLS são suficientes para verificar a resposta do OCSP.
- A resposta do OCSP pode ser assinada digitalmente usando outro certificado assinado pela mesma autoridade de certificação (CA) que emitiu o certificado que você está verificando. O certificado de assinatura é enviado junto com a resposta do OCSP nesse caso. O certificado que fluiu do respondente OCSP deve ter uma Extensão de Uso de Chave Estendida configurada como `id-kp-OCSPSigning` para que possa ser confiável para este propósito. Como a resposta do OCSP é enviada com o certificado que a assinou (e esse certificado é assinado por uma CA já confiável para a conectividade do TLS), nenhuma configuração de certificado adicional é necessária.
- A resposta do OCSP pode ser assinada digitalmente usando outro certificado que não esteja relacionado diretamente ao certificado que estiver verificando. Neste caso, a resposta do OCSP é assinada por um certificado emitido pelo próprio respondente do OCSP. Deve-se incluir uma cópia do certificado do respondente OCSP para o banco de dados de chaves do cliente ou do gerenciador de filas que executa a verificação de OCSP. Consulte o “Incluindo um certificado de autoridade de certificação ou a parte pública de um certificado confiável em um repositório de chaves no AIX, Linux, and Windows” na página 559. Quando um certificado de CA é incluído, por padrão, ele é incluído como uma raiz confiável, que é a configuração necessária nesse contexto. Se este certificado não for incluído, o IBM MQ não poderá verificar a assinatura digital na resposta do OCSP e a verificação do OCSP resultará em um resultado Desconhecido, que pode fazer com que o IBM MQ feche o canal, dependendo do valor de OCSPAuthentication.

Online Certificate Status Protocol (OCSP) em aplicativos cliente Java e JMS

Devido a uma limitação da API Java, o IBM MQ pode usar a verificação de revogação de certificados do Online Certificate Status Protocol (OCSP) para soquetes seguros TLS somente quando o OCSP está ativado para o todo o processo de Java virtual machine (JVM). Existem duas maneiras de ativar o OCSP para todos os soquetes seguros na JVM:

- Edite o arquivo JRE java.security para incluir as definições de configuração do OCSP mostradas na Tabela 1 e reinicie o aplicativo.
- Use a API java.security.Security.setProperty(), sujeito a qualquer política do Java Security Manager em vigor.

No mínimo, é necessário especificar um dos valores ocsp.enable e ocsp.responderURL.

Nome da propriedade	Descrição
ocsp.enable	Este valor da propriedade é `true` ou `false`. Se `true`, a verificação de OCSP é ativada ao realizar a verificação de revogação de

Nome da propriedade	Descrição
	certificados; se `false` ou não configurado, a verificação de OCSP está desativada.
ocsp.responderURL	Este valor de propriedade é uma URL que identifica o local do respondente do OCSP. Aqui está um exemplo; `ocsp.responderURL=http://ocsp.example.net:80`. Por padrão, o local do OCSP respondente é determinado implicitamente a partir do certificado que estiver sendo validado. A propriedade será usada quando a extensão Acesso de Informações de Autoridade (definida na RFC 3280) estiver ausente do certificado ou quando requerer substituição.
ocsp.responderCertSubjectName	Este valor da propriedade é o nome do assunto do certificado do respondente do OCSP. Aqui está um exemplo; `ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"`. Por padrão, o certificado do respondente do OCSP é aquele do emissor do certificado que está sendo validado. Esta propriedade identifica o certificado do respondente do OCSP quando o padrão não se aplica. Seu valor é um nome distinto de sequência (definido no RFC 2253) que identifica um certificado no conjunto de certificados que são fornecidos durante a validação do caminho do certificado. Nos casos em que o nome do assunto sozinho não é suficiente para identificar exclusivamente o certificado, as propriedades ocsp.responderCertIssuerName e ocsp.responderCertSerialNumber devem ser usadas. Quando esta propriedade for definida, então as propriedades ocsp.responderCertIssuerName e ocsp.responderCertSerialNumber serão ignoradas.
ocsp.responderCertIssuerName	Este valor da propriedade é o nome do emissor do certificado do respondente do OCSP. Aqui está um exemplo; `ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"`. Por padrão, o certificado do respondente do OCSP é aquele do emissor do certificado que está sendo validado. Esta propriedade identifica o certificado do respondente do OCSP quando o padrão não se aplica. Seu valor é um nome distinto de sequência (definido no RFC 2253) que identifica um certificado no conjunto de certificados que são fornecidos durante a validação do caminho do certificado. Quando esta propriedade for definida, a propriedade ocsp.responderCertSerialNumber também deve ser configurada. Essa propriedade é ignorada quando a propriedade ocsp.responderCertSubjectName está configurada.
ocsp.responderCertSerialNumber	Este valor da propriedade é o número de série do certificado do respondente do OCSP. Aqui está um exemplo; `ocsp.responderCertSerialNumber=2A:FF:00`. Por padrão, o certificado do respondente do OCSP é aquele do emissor do certificado que está sendo validado. Esta propriedade identifica o certificado do respondente do OCSP quando o padrão não se aplica. Este valor é uma sequência de dígitos hexadecimais (separadores de dois pontos ou espaço podem estar presentes) que identifica um certificado no conjunto dos certificados fornecidos durante a validação do caminho do certificado. Quando esta propriedade for definida, a propriedade ocsp.responderCertIssuerName também deve ser configurada. Essa propriedade é ignorada quando a propriedade ocsp.responderCertSubjectName está configurada.

Antes de ativar do OCSP dessa forma, há várias considerações:

- Definir a configuração do OCSP afeta todos os soquetes seguros no processo da JVM. Em alguns casos, essa configuração pode ter efeitos colaterais indesejáveis quando a JVM é compartilhada com outro código do aplicativo que usa soquetes seguros TLS. Certifique-se de que a configuração do OCSP escolhido é adequada para todos os aplicativos que estão em execução na mesma JVM.
- Aplicar a manutenção para o seu JRE poderá sobrescrever o arquivo java.security. Tome cuidado ao aplicar correções temporárias do Java e manutenção do produto para evitar sobrescrever o arquivo java.security. Pode ser necessário reaplicar suas alterações java.security depois de aplicar a manutenção. Por essa razão, considere definir a configuração do OCSP usando a API java.security.Security.setProperty().
- Ativar a verificação de OCSP terá efeito somente se a verificação de revogação também estiver ativada. A verificação de revogação é ativada pelo método `PKIXParameters.setRevocationEnabled()`.
- Se você estiver usando o AMS Java Interceptor descrito em Ativando verificação de OCSP em interceptores nativos, tome cuidado para evitar o uso de uma configuração do OCSP java.security que entra em conflito com a configuração do OCSP AMS no arquivo de configuração de keystore.

Trabalhando com as Listas de Revogação de Certificados e Listas de Revogação de Autoridade

O suporte do IBM MQ para CRLs e ARLs varia por plataforma.

O suporte CRL e ARL em cada plataforma é o seguinte:

- Multi Em Multiplataformas, o suporte CRL e ARL está em conformidade com as recomendações do perfil CRL PKIX X.509 V2 .
- z/OS No z/OS, o SSL do Sistema suporta as CRLs e ARLs armazenadas nos servidores LDAP pelo produto Tivoli Public Key Infrastructure.

IBM MQ mantém um cache de CRLs e ARLs que foram acessadas nas últimas 12 horas.

Quando um gerenciador de filas ou IBM MQ MQI client recebe um certificado, ele verifica a CRL para confirmar se o certificado ainda é válido. O IBM MQ primeiro verifica no cache, se houver um cache. Se o CRL não estiver no cache, o IBM MQ interrogará os locais do servidor LDAP CRL na ordem em que eles ocorrem na lista de nomes de objetos de informações sobre autenticação especificados pelo atributo *SSLCRLNL*, até que o IBM MQ localize um CRL disponível. Se a lista de nomes não estiver especificada ou está especificada com um valor em branco, as CRLs não são verificadas.

Configurando Servidores LDAP

Configure a Estrutura em Árvore de Informações do Diretório LDAP para que reflita a hierarquia de Nomes Distintos de CAs. Faça isso usando arquivos de Formato de Troca de Dados LDAP.

Configure a estrutura do LDAP DIT (Directory Information Tree) para utilizar a hierarquia correspondente aos Nomes Distintos das CAs que emitem certificados e CRLs. Você pode definir a estrutura DIT com um arquivo que utiliza o LDAP LDIF (Data Interchange Format). Você também pode utilizar arquivos do LDIF para atualizar um diretório.

Arquivos LDIF são arquivos de texto ASCII que contém as informações exigidas para definir objetos dentro do diretório LDAP. Os arquivos LDIF contêm uma ou mais entradas, cada uma das quais compreende um Nome Distinto, pelo menos uma definição de classe de objeto e, como opção, várias definições de atributos.

O atributo `certificateRevocationList;binary` contém uma lista em formato binário de certificados de usuários revogados. O atributo `authorityRevocationList;binary` contém uma lista binária de certificados CA que foram revogados. Para uso com o TLS do IBM MQ, os dados binários para esses atributos devem em estar conformidade com o formato DER (Regras Distintas de Codificação). Para obter mais informações sobre os arquivos LDIF, consulte a documentação fornecida com o seu servidor LDAP.

A Figura 20 na página 351 mostra um arquivo LDIF de amostra que pode ser criado como entrada para o servidor LDAP carregar as CRLs (listas de revogação de certificado) e ARLs emitidas pela CA1, que é uma Autoridade de certificação imaginária com o Nome distinto "CN=CA1, OU=Test, O=IBM, C=GB", configurado pela Organização de teste na IBM.

```
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)
```

Figura 20. Arquivo LDIF de amostra para uma Autoridade de certificação. Isso poderá variar de implementação para implementação.

Figura 21 na página 351 mostra a estrutura DIT que o seu servidor LDAP cria ao carregar o arquivo LDIF de amostra mostrado no Figura 20 na página 351 juntamente com um arquivo semelhante para a CA2, uma Autoridade de certificação imaginária definida pela organização PKI, também dentro do IBM.

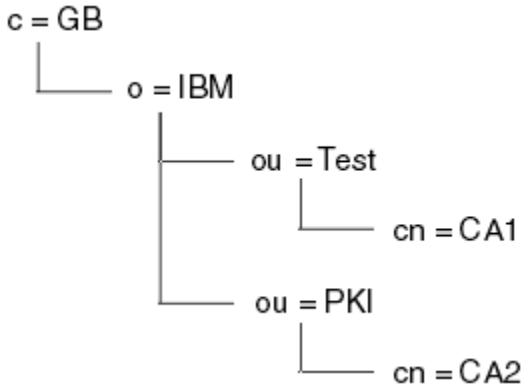


Figura 21. Exemplo de estrutura de Directory Information Tree do LDAP

O IBM MQ verifica ambos, CRLs e ARLs.

Nota: Certifique-se de que a lista de controle de acesso para seu servidor LDAP permita que usuários autorizados leiam, pesquisem e comparem as entradas que contenham as CRLs e ARLs. O IBM MQ acessa o servidor LDAP usando as propriedades LDAPUSER e LDAPPWD do objeto AUTHINFO

Configurando e Atualizando Servidores LDAP
Use este procedimento para configurar ou atualizar o servidor LDAP.

1. Obtenha as CRLs e ARLs em formato DER a partir da sua Autoridade ou Autoridades de Certificação.
2. Utilizando o editor de texto ou a ferramenta fornecida em seu servidor LDAP, crie um ou mais arquivos LDIF que contenham o Nome Distinto da CA e as definições da classe de objetos exigidas. Copie os dados do formato DER para o arquivo LDIF como os valores do atributo `certificateRevocationList;binary` atributo para CRLs, o atributo `authorityRevocationList;binary` para ARLs ou ambos.
3. Inicie seu servidor LDAP.

4. Inclua as entradas do arquivo LDIF ou arquivos que você criou na etapa “2” na página 351.

Depois de configurar o servidor LDAP CRL, verifique se ele está configurado corretamente. Primeiro, tente utilizar um certificado que não esteja revogado no canal, e verifique se o canal foi iniciado corretamente. Em seguida utilize um certificado revogado e verifique se o canal falhou ao iniciar.

Obtenha regularmente as CRLs atualizadas a partir das Autoridades de certificação. Tente fazer esta operação em seus servidores LDAP a cada 12 horas.

Acessando as CRLs e ARLs com um Gerenciador de Filas

Um gerenciador de filas é associado a um ou mais objetos de informação de autenticação, que contêm o endereço de um servidor de LDAP CRL. IBM i IBM MQ on IBM i se comporta de forma diferente de outras plataformas.

Observe que nesta seção, as informações sobre as CRLs (Certificate Revocation Lists) também se aplicam às ARLs (Authority Revocation Lists).

Informe ao gerenciador de filas como acessar as CRLs ao fornecê-los os objetos de informações de autenticação, cada qual mantendo o endereço de um servidor de CRL LDAP. Os objetos de informações sobre autenticação são mantidos em um lista de nomes, que é especificada no atributo de gerenciador de filas *SSLCRLNL*.

No seguinte exemplo, o MQSC é utilizado para especificar os parâmetros:

1. Defina os objetos de informações de autenticação usando o comando DEFINE AUTHINFO MQSC com o parâmetro AUTHTYPE definido para CRLLDAP. IBM i No IBM i, também é possível o comando CRTMQMAUTI CL.

   O valor CRLLDAP para o parâmetro AUTHTYPE indica que as CRLs são acessadas em servidores LDAP. Cada objeto de informações de autenticação com o tipo CRLLDAP que você criar contém o endereço de um servidor LDAP. Quando você tem mais de um objeto de informações sobre autenticação, os servidores LDAP para os quais eles apontam devem conter informações idênticas. Isso fornece continuidade de serviço caso ocorra uma ou mais falhas nos servidores LDAP.

   z/OS Além disso, somente no z/OS, todos os servidores LDAP devem ser acessados usando o mesmo ID do usuário e senha. O ID do usuário e senha utilizados são aqueles especificados no primeiro objeto AUTHINFO na lista de nomes.

   Em todas as plataformas, o ID do usuário e a senha são enviados não criptografados para o servidor LDAP.
2. Utilizando o comando DEFINE NAMELIST MQSC, defina uma lista de nomes para os nomes de seus objetos de informações de autenticação. z/OS No z/OS, assegure-se de que o atributo da lista de nomes NLTYPE esteja configurado como AUTHINFO.
3. Utilizando o comando ALTER QMGR MQSC, forneça a lista de nomes ao gerenciador de filas. Por exemplo:

   ```
   ALTER QMGR SSLCRLNL(sslcrlnlname)
   ```

   em que `sslcrlnlname` é a sua lista de nomes de objetos de informações sobre autenticação

   Esse comando configura um atributo de gerenciador de filas chamado *SSLCRLNL*. O valor inicial do gerenciador de filas para este atributo está em branco.

IBM i No IBM i, é possível especificar os objetos de informações de autenticação, porém o gerenciador de filas não usa nem os objetos de informações de autenticação, nem uma lista de nomes de objetos de informações de autenticação. Apenas os clientes do IBM MQ que usam uma tabela de conexão de cliente gerada por um Gerenciador de Filas do IBM i usam as informações de autenticação especificadas para esse Gerenciador de Filas do IBM i. O atributo de gerenciador de filas *SSLCRLNL* no IBM i determina quais informações sobre autenticação são usadas por esses clientes. Consulte

“Acessando CRLs e ARLs no IBM i” na página 353 para obter informações sobre como informar um gerenciador de filas do IBM i como acessar as CRLs.

É possível incluir até 10 conexões para servidores LDAP alternativos para a lista de nomes, para garantir a continuidade do serviço se um ou mais servidores LDAP falharem. Observe que os servidores LDAP devem conter informações idênticas.

IBM i *Acessando CRLs e ARLs no IBM i*

Use este procedimento para acessar CRLs ou ARLs no IBM i.

Observe que nesta seção, as informações sobre as CRLs (Certificate Revocation Lists) também se aplicam às ARLs (Authority Revocation Lists).

Siga estas etapas para configurar um local de CRL para um certificado específico no IBM i:

1. Acesse a interface DCM, conforme descrito em “Acessando DCM” na página 280.
2. Na categoria de tarefas **Gerenciar locais de CRL** no painel de navegação, clique em **Incluir local de CRL**. A página Gerenciar Locais de CRL é exibida no quadro de tarefas.
3. No campo **Nome do Local da CRL** , digite um nome do local da CRL, por exemplo LDAP `Server #1`
4. No campo **Servidor LDAP**, digite o nome do servidor LDAP.
5. No campo **Usar Secure Sockets Layer (SSL)**, selecione **Sim** se desejar conectar-se ao servidor LDAP usando TLS. Caso contrário, selecione **Não**.
6. No campo **Número da Porta**, digite um número da porta para o servidor LDAP, por exemplo, 389.
7. Se o seu servidor LDAP não permitir que usuários anônimos acessem o diretório, digite um nome distinto de login para o servidor no campo **Nome Distinto de Login**.
8. Clique em **OK**. O DCM informará que o local da CRL foi criado.
9. No painel de navegação, clique em **Selecionar um Armazenamento de Certificados**. A página Selecionar um Armazenamento de Certificados é exibida no quadro de tarefas.
10. Selecione a caixa de opção **Outro Armazenamento de Certificados do Sistema** e clique em **Continuar**. A página Armazenamento de Certificados e Senha é exibida.
11. No campo **Caminho e nome de arquivo de armazenamento de certificados**, digite o caminho e nome de arquivo IFS configurado quando “Criando um armazenamento de certificados no IBM i” na página 283.
12. Digite uma senha no campo **Senha do Armazenamento de Certificados**. Clique em **Continue**. A página Armazenamento de Certificados Atual é exibida no quadro de tarefas.
13. Na categoria de tarefas **Gerenciar Certificados** no painel de navegação, clique em **Atualizar Designação do Local da CRL**. A página Designação do Local de CRL é exibida no quadro de tarefas.
14. Selecione o botão de opção para o certificado de CA para o qual deseja atribuir o local da CRL. Clique em **Atualizar a Designação do Local da CRL**. A página Atualizar Designação do Local de CRL é exibida no quadro de tarefas.
15. Selecione o botão de opção para o local de CRL para o qual deseja atribuir o certificado. Clique em **Atualizar Designação**. O DCM informará que a designação foi atualizada

Observe que o DCM permite que você atribua um servidor LDAP diferente pela Autoridade de certificação.

Acessando CRLs e ARLs usando o IBM MQ Explorer

É possível usar o IBM MQ Explorer para informar a um gerenciador de filas como acessar CRLs.

Observe que nesta seção, as informações sobre as CRLs (Certificate Revocation Lists) também se aplicam às ARLs (Authority Revocation Lists).

Utilize o seguinte procedimento para definir uma conexão LDAP para uma CRL:

1. Certifique-se de que iniciou o gerenciador de filas.
2. Clique com o botão direito na pasta **Informações sobre autenticação** e clique em **Novo -> Informações sobre autenticação**. Na folha de propriedade que abre:

a. Na primeira página **Criar Informações de Autenticação**, digite um nome para o objeto CRL (LDAP).
b. Na página **Geral** de **Alterar Propriedades**, selecione o tipo de conexão. Opcionalmente é possível digitar uma descrição.
c. Selecione a página **CRL(LDAP)** de **Alterar Propriedades**.
d. Digite o nome do servidor LDAP como o nome da rede ou o endereço IP.
e. Caso o servidor exija os detalhes de login, forneça o ID do usuário e, se for preciso, uma senha.
f. Clique em **OK**.

3. Clique com o botão direito na pasta Lista de nomes e clique em **Nova-> Lista de nomes**. Na folha de propriedade que abre:
   a. Digite um nome para a lista de nomes.
   b. Inclua o nome do objeto CRL(LDAP) (da etapa "2.a" na página 354) na lista.
   c. Clique em **OK**.
4. Clique com o botão direito do mouse no gerenciador de filas, selecione **Propriedades** e selecione a página **SSL**:
   a. Selecione a caixa de entrada **Verificar certificados recebidos por este gerenciador de filas comparandos às Listas de Certificados Revogados**.
   b. Digite o nome da lista de nomes (da etapa "3.a" na página 354) no campo **Nomes da CRL**.

Acessando as CRLs e ARLs com um IBM MQ MQI client

Há três opções para especificar os servidores LDAP que contêm CRLs para verificar por um IBM MQ MQI client.

Observe que nesta seção, as informações sobre as CRLs (Certificate Revocation Lists) também se aplicam às ARLs (Authority Revocation Lists).

As três maneiras de especificar os servidores LDAP são as seguintes:

- Utilizando uma tabela de definição de canais
- Usando a estrutura de opções de configuração do SSL, MQSCO, ou uma chamada de MQCONNX
- Usando o Active Directory (em sistemas Windows com suporte ao Active Directory)

Para obter mais detalhes, consulte as informações relacionadas.

É possível incluir até 10 conexões para servidores LDAP alternativos, para garantir a continuidade do serviço se um ou mais servidores LDAP falharem. Observe que os servidores LDAP devem conter informações idênticas.

Não é possível acessar CRLs do LDAP a partir de um canal IBM MQ MQI client em execução no Linux (plataforma zSeries).

Local de um respondente OCSP e dos servidores LDAP que contêm CRLs
Em um sistema IBM MQ MQI client,é possível especificar o local de um respondente OCSP e dos servidores Lightweight Directory Access Protocol (LDAP) que contêm listas de revogação de certificado (CRLs).

É possível especificar esses locais de três maneiras descritas aqui, em ordem de precedência decrescente.

IBM i Para o IBM i, consulte Acessando CRLs e ARLs no IBM i.

Quando um aplicativo IBM MQ MQI client emite uma chamada MQCONNX

É possível especificar um respondente OCSP ou um servidor LDAP que contém CRLs em uma chamada **MQCONNX**.

Em uma chamada **MQCONNX**, a estrutura de opções de conexão, MQCNO, pode referenciar uma estrutura de opções de configuração de SSL, MQSCO. Por sua vez, a estrutura MQSCO pode referenciar uma ou

mais estruturas de registro de informações sobre autenticação, MQAIR. Cada estrutura MQAIR contém todas as informações que um IBM MQ MQI client requer para acessar um respondente OCSP ou um servidor LDAP que contém CRLs. Por exemplo, um dos campos em uma estrutura MQAIR é a URL na qual um replicador pode ser contatado. Para obter mais informações sobre a estrutura MQAIR, consulte MQAIR - Registro de informações sobre autenticação.

Usando uma tabela de definição de canal de cliente (ccdt) para acessar um respondente OCSP ou servidores LDAP

Para que um IBM MQ MQI client pode acessar um respondente OCSP ou servidores LDAP que contêm CRLs, incluem os atributos de um ou mais objetos de informação de autenticação em uma tabela de definição de canal do cliente.

Em um gerenciador de filas do servidor é possível definir um ou mais objetos de informações sobre autenticação. Os atributos de um objeto de autenticação contêm todas as informações que são necessárias para acessar um respondente OCSP (em plataformas em que OCSP é suportado) ou um servidor LDAP que contém CRLs. Um dos atributos especifica a URL do respondente OCSP, outro especifica o endereço do host ou endereço IP de um sistema no qual um servidor LDAP é executado.

z/OS IBM i Um objeto de informação de autenticação com AUTHTYPE(OCSP) não se aplica para uso no IBM i ou z/OS os gerenciadores de filas, mas pode ser especificado nas plataformas a serem copiadas na tabela de definição de canal do cliente (CCDT) para uso do cliente.

Para ativar um IBM MQ MQI client para acessar um respondente OCSP ou servidores LDAP que contêm CRLs, os atributos de um ou mais objetos de informação de autenticação podem ser incluídos em uma tabela de definição de canal do cliente. É possível incluir atributos em uma das seguintes maneiras:

Multi

Nas plataformas do servidor AIX, Linux, IBM i e Windows

É possível definir uma lista de nomes que contém os nomes de um ou mais objetos de informações de autenticação. Em seguida, é possível configurar o atributo de gerenciador de filas, **SSLCRLNL**, para o nome dessa lista de nomes.

Se você estiver usando CRLs, mais de um servidor LDAP poderá ser configurado para fornecer maior disponibilidade. A intenção é que cada servidor LDAP contenha as mesmas CRLs. Se um servidor LDAP estiver indisponível quando for necessário, um IBM MQ MQI client poderá tentar acessar outro.

Os atributos dos objetos de informação de autenticação identificados pela lista de nomes são referidos coletivamente aqui como o *local de revogação de certificado*. Ao configurar o atributo de gerenciador de filas, **SSLCRLNL**, para o nome da lista de nomes, o local de revogação de certificado é copiado para a tabela de definição de canal de cliente associada ao gerenciador de filas. Se a CCDT puder ser acessada a partir de um sistema do cliente como um arquivo compartilhado ou se a CCDT for então copiada em um sistema do cliente, o IBM MQ MQI client nesse sistema poderá usar o local de revogação de certificado na CCDT para acessar um replicador OCSP ou servidores LDAP que contêm CRLs.

Se o local de revogação de certificado do gerenciador de filas for mudado posteriormente, a mudança será refletida na CCDT associada ao gerenciador de filas. Se o atributo de gerenciador de filas, **SSLCRLNL**, for configurado em branco, o local de revogação de certificado será removido do CCDT. Estas alterações não são refletidas em nenhuma cópia da tabela em um sistema do cliente.

Se você requerer que o local de revogação de certificado nas extremidades do cliente e do servidor de um canal MQI seja diferente, e o gerenciador de filas do servidor for aquele que é usado para criar o local de revogação de certificado, será possível fazer isto conforme a seguir:

1. No gerenciador de filas do servidor, crie o local de revogação de certificado para uso no sistema do cliente.
2. Copie a CCDT contendo o local de revogação de certificado no sistema do cliente.

3. No gerenciador de filas do servidor, altere o local de revogação de certificado para o que é necessário na extremidade do servidor do canal MQI.
4. Na máquina cliente, é possível usar o comando **runmqsc** com o parâmetro **-n**.

Multi

Em plataformas do cliente AIX, Linux, IBM i e Windows

É possível compilar um CCDT na máquina cliente usando o comando runmqsc com o parâmetro **-n** e objetos **DEFINE AUTHINFO** no arquivo CCDT. A ordem em que os objetos são definidos em é a ordem na qual eles são usados no arquivo. Qualquer nome que você possa usar em um objeto **DEFINE AUTHINFO** não será retido no arquivo. Somente números posicionais são usados quando você **DISPLAY** os objetos **AUTHINFO** em um arquivo CCDT.

Nota: Se o parâmetro **-n** for especificado, não se deve especificar qualquer outro parâmetro.

Usando o Active Directory no Windows

Windows

Nos sistemas Windows é possível usar o comando de controle **setmqcrl** para publicar as informações de CRL atuais no Active Directory.

O comando **setmqcrl** não publica informações de OCSP.

Para obter informações sobre este comando e sua sintaxe, consulte setmqcrl.

Acessando as CRLs e ARLs com o IBM MQ classes for Java e IBM MQ classes for JMS

O IBM MQ classes for Java e o IBM MQ classes for JMS acessam as CRLs de forma diferente de outras plataformas.

Para obter informações sobre como trabalhar com CRLs e ARLs com o IBM MQ classes for Java, consulte Usando listas de revogação de certificado

Para obter informações sobre como trabalhar com CRLs e ARLs com IBM MQ classes for JMS, consulte Propriedade de objeto SSLCERTSTORES

Manipulando Objetos de Informações de Autenticação

É possível manipular objetos de informações sobre autenticação usando os comandos MQSC ou PCF ou o IBM MQ Explorer.

Os seguintes comandos MQSC agem sobre objetos informações de autenticação:

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

Para uma descrição completa desses comandos, veja Comandos MQSC.

Os seguintes comandos Programmable Command Format (PCF) agem sobre objetos de informações de autenticação:

- Criar Informações sobre Autenticação
- Copiar Informações sobre Autenticação
- Alterar Informações sobre Autenticação
- Excluir Informações sobre Autenticação
- Consultar Informações sobre Autenticação
- Consultar Nomes de Informações sobre Autenticação

Para obter uma descrição completa desses comandos, consulte Definições dos formatos de comando programáveis.

Em plataformas em que ele estiver disponível, também será possível usar o IBM MQ Explorer.

Linux AIX Usando o Pluggable Authentication Method (PAM)

É possível usar o PAM apenas em plataformas AIX and Linux. Um sistema típico AIX ou Linux tem módulos PAM que implementam o mecanismo de autenticação tradicional; no entanto, pode haver mais. Assim como a tarefa básica de validação de senhas, os módulos PAM também podem ser chamados para realizar regras adicionais.

Os arquivos de configuração definem qual método de autenticação deve ser usado para cada aplicativo. Os aplicativos de exemplo incluem o login do terminal padrão, ftp e telnet.

A vantagem do PAM é que o aplicativo não precisa saber ou se importar sobre como o ID do usuário está realmente sendo autenticado. Desde que o aplicativo possa fornecer uma formulário correto de dados de autenticação para PAM, o mecanismo por trás dele é transparente.

O formulário de dados de autenticação depende do sistema que está sendo usado. Por exemplo, o IBM MQ obtém uma senha por meio de parâmetros, como a estrutura MQCSP usada na chamada de API `MQCONNX`.

Importante: Não é possível configurar o atributo **AUTHENMD** até que você instale o IBM MQ 8.0.0 Fix Pack 3 e, em seguida, reinicie o gerenciador de filas, usando um **-e CMDLEVEL=** *level* de *802* (no comando strmqm) para configurar o nível de comando necessário.

Configurando seu sistema para usar o PAM

O nome do serviço usado pelo IBM MQ ao chamar o PAM é *ibmmq*.

Observe que uma instalação do IBM MQ tenta manter uma configuração de PAM padrão que permite conexões de usuários do sistema operacional com base nos padrões conhecidos para os sistemas operacionais diferentes.

No entanto, seu administrador do sistema deve verificar se nas regras definidas no `/etc/pam.conf` ou no `/etc/pam.d/ibmmq`, os arquivos ainda são apropriados.

Autorizando o acesso aos objetos

Esta seção contém informações sobre como usar o gerenciador de autoridade de objeto e programas de saída de canal para controlar o acesso aos objetos.

ALW Em sistemas AIX, Linux, and Windows. o acesso a objetos é controlado usando o gerenciador de autoridade de objeto (OAM). Esta coleção de tópicos contém informações sobre como usar a interface de comando para o OAM.

Esta seção também contém uma lista de verificação que é possível usar para determinar quais tarefas executar para aplicar segurança em seu sistema em todas as plataformas e as considerações para conceder aos usuários a autoridade para administrar o IBM MQ e trabalhar com os objetos do IBM MQ.

Se os mecanismos de segurança fornecidas não atenderem suas necessidades, é possível desenvolver seus próprios programas de saída de canal.

Determinando qual usuário é usado para autorização

As autoridades para acessar recursos são concedidas a grupos dos quais o usuário é membro ou, em determinados modos, diretamente ao usuário associado à conexão.. Durante o processo de conexão, e em particular para conexões remotas (cliente), essa identidade poderia ser alterada pela configuração do gerenciador de filas. Esta página lista os diferentes recursos do IBM MQ e suas opções de configuração que podem impactar a identidade de um aplicativo de conexão e a ordem de precedência na qual esses recursos entram em vigor

Recursos que podem modificar qual usuário é adotado

Os diferentes recursos que podem configurar qual usuário deve ser autorizado são os seguintes:

Usuário declarado do aplicativo
Quando uma conexão remota é iniciada pelo IBM MQ, o usuário do sistema operacional com o qual o processo está em execução é enviado para o gerenciador de filas de recebimento Esse usuário é enviado para assegurar que, se não existir nenhuma configuração adicional que modifique o usuário, haja um usuário que possa ser usado para verificação de autorização

Não é recomendado usar esse usuário como base para autorização, pois ele permite que as conexões declarem sua identidade sem qualquer validação do lado do servidor. Isso pode até incluir o usuário administrativo ('mqm ').

Configuração MCAUSER do Canal
Aplicativos que se conectam por meio de ligações de rede fazem isso usando uma definição de canal IBM MQ . As definições de canal suportam o atributo **MCAUSER** , que pode ser usado para especificar um usuário diferente a ser usado para autorização em vez do usuário declarado pelos aplicativos de conexão.

Autenticação de conexão ADOPTCTX
Os aplicativos podem especificar um usuário e uma senha a serem enviados para um gerenciador de filas para propósitos de autenticação Essas credenciais são autenticadas usando a configuração especificada para o recurso Autenticação de Conexão. A opção **ADOPTCTX** para Autenticação de Conexão controla se um usuário deve ser usado para autorização após ele ter sido validado com êxito Se configurado como YES, o usuário fornecido para autenticação será adotado para verificações de autorização.

V 9.4.0 No IBM MQ 9.3.4, um token pode ser fornecido para autenticação, se **ADOPTCTX** for configurado como YES, um usuário será adotado a partir das solicitações que o token contém.

Registro de autenticação de canal MCAUSER
Durante o processamento da conexão, o gerenciador de filas tentará localizar um registro de autenticação de canal correspondente à conexão. Se um registro de autenticação de canal for correspondido e seu valor de atributo **USERSRC** for configurado como MAP, IBM MQ mudará o usuário usado para autorizações para o valor do atributo **MCAUSER** .

Saídas de Segurança
Saídas de segurança são funções customizadas que podem ser gravadas e chamadas durante o processamento de segurança IBM MQ . Quando a função é chamada, ela é fornecida com uma cópia da estrutura MQCD que inclui vários campos relacionados ao usuário de conexões que serão usados para verificações de autorização... As saídas de segurança podem modificar esses campos para alterar o usuário que será autorizado

ordem de precedência

A tabela a seguir mostra a ordem de precedência para cada recurso de segurança descrito no "Recursos que podem modificar qual usuário é adotado" na página 358 quando o IBM MQ está selecionando um usuário para autorizar A ordem é do mais baixo para o mais alto, ou seja, um recurso de segurança que define um usuário na primeira linha é substituído por qualquer uma das outras linhas.

Tabela 71. Ordem de precedência para recursos de segurança

Ordem	**Recurso**
1 (mais baixo)	ID declarado do aplicativo
2	Atributo **MCAUSER** de definição de canal
3	Autenticação de conexão com **ADOPTCTX(YES)**
4	Registros de autenticação de canal com **USERSRC(MAP)**

Tabela 71. Ordem de precedência para recursos de segurança (continuação)	
Ordem	**Recurso**
5 (mais alto)	Saída de segurança

Implicações da adoção precoce

A autenticação de conexão e registros de autenticação de canal fornecem uma opção de configuração que controla quando a adoção do usuário de autenticação de conexão é executada. Essa configuração é referida como adoção antecipada.. Se a adoção antecipada estiver ativada, a adoção da identidade de autenticação de conexão ocorrerá antes que os registros de autenticação de canal sejam processados (o que significa que os registros de autenticação de canal substituem qualquer adoção do **CONNAUTH** .

Se desativada, a ordem será revertida-ou seja, os registros de autenticação de canal serão processados antes da adoção do **CONNAUTH** Nessa situação, a adoção da autenticação de conexão tem uma prioridade efetiva mais alta que a autenticação de canal.

A configuração padrão para adoção antecipada é `enabled`..

ALW Controlando o acesso a objetos usando o OAM no AIX, Linux, and Windows

O gerenciador de autoridade de objeto (OAM) fornece uma interface de comando para conceder e revogar autoridade para objetos do IBM MQ.

Você deve estar adequadamente autorizado para usar esses comandos, conforme descrito em "Autoridade para administrar o IBM MQ no AIX, Linux, and Windows" na página 407. Os IDs de usuários que estão autorizados a administrar o IBM MQ, tem autoridade de *super usuário* para o gerenciador de filas, o que significa que não é necessário lhes conceder permissão adicional para emitir quaisquer pedidos ou comandos MQI.

Linux AIX Permissões baseadas em usuário do OAM no AIX and Linux

Em sistemas UNIX and Linux, o gerenciador de autoridade de objeto (OAM) pode usar a autorização baseada em usuário, assim como a autorização baseada em grupo.

Antes da IBM MQ 8.0, as listas de controle de acesso (ACLs) no UNIX and Linux eram baseadas apenas em grupos. DeIBM MQ 8.0 , as ACLs são baseadas em IDs de usuário e grupos, e você pode usar o modelo baseado em usuário ou o modelo baseado em grupo para autorização definindo a propriedade**SecurityPolicy** atributo ao valor apropriado conforme descrito em Estrofe de serviço doqm.ini arquivo .

Mudanças no comportamento da IBM MQ 8.0 e mais recente

A partir da IBM MQ 8.0, ao executar com a política baseada em usuário, alguns comandos retornam informações diferentes de versões anteriores do produto:

- Os comandos **dmpmqaut** e **dmpmqcfg** mostram registros baseados em usuário, como fazem as operações equivalentes do PCF.
- O plug-in do OAM para o IBM MQ Explorer mostra registros baseados em usuário e permite modificações baseadas em usuário.
- A função **Inquire** do OAM retorna resultados que mostram que ela é compatível com o usuário.

Usar o atributo **-p** no comando **setmqaut** não concede acesso a todos os usuários no mesmo grupo primário, quando as autorizações baseadas em usuário são ativadas no arquivo `qm.ini` , conforme descrito na sub-rotina Serviço do arquivo qm.ini.

Se você começar a empregar a autorização baseada em usuário e tiver muitos usuários, provavelmente haverá mais registros que são armazenados na fila AUTH do que com o modelo baseado em grupo, e o processo de autorização poderá demorar um pouco mais do que antes pois há mais registros a serem verificados. Não espera-se que este aumento seja significativo. Se necessário, é possível usar uma mistura de permissões de usuário e de grupo.

Considerações Sobre Migração

Se você alterar o modelo de grupo para usuário para um gerenciador de filas existente, não haverá efeito imediato. As autorizações que já foram feitas continuam a se aplicar. Qualquer usuário que se conecta ao gerenciador de filas recebe os mesmos privilégios que antes: a combinação de todos os grupos aos quais seu ID pertence. Quando novos comandos **setmqaut** forem emitidos para IDs do usuário, eles terão efeito imediato.

Se você criar um novo gerenciador de filas com a política do usuário, esse gerenciador de filas terá permissões apenas para o usuário que o criar (que é normalmente, mas não necessariamente, o ID do usuário `mqm`). Há também permissões que são concedidas automaticamente ao grupo `mqm`. No entanto, se você não tiver o `mqm` como o grupo primário, então o grupo `mqm` não será incluído no conjunto inicial de autorizações.

Se você mudar de uma política de usuário para grupo, as autorizações baseadas em usuário não serão automaticamente excluídas. No entanto, elas não serão mais usadas durante a verificação de permissões. Antes de reverter a política, salve a configuração atual, altere a política, reinicie o gerenciador de filas e, em seguida, reproduza o script. Como agora é um gerenciador de filas baseado em grupo, o efeito é que as regras desse ID do usuário serão armazenadas com base no grupo primário.

Conceitos relacionados

Gerenciador de autoridade de objeto (OAM)

“Principais e grupos no AIX, Linux, and Windows” na página 412

Principais podem pertencer a grupos. Concedendo acesso de recurso a grupos em vez de indivíduos, é possível reduzir a quantidade de administração requeridos. As Listas de controle de acesso (ACLs) baseiam-se em ambos os grupos e IDs de usuário.

Referências relacionadas

Sub-rotina Service do arquivo qm.ini

Comando **crtmqm** (criar gerenciador de filas)

ALW Concedendo acesso a um objeto do IBM MQ no AIX, Linux, and Windows

Use o comando de controle **setmqaut**, o comando MQSC **SET AUTHREC** ou o comando PCF **MQCMD_SET_AUTH_REC** para fornecer aos usuários e grupos de usuários o acesso aos objetos do IBM MQ. Observe que no IBM MQ Appliance é possível usar apenas o comando **SET AUTHREC** .

Para obter uma definição completa do comando de controle **setmqaut** e sua sintaxe, consulte setmqaut.

Para obter uma definição completa do comando MQSC **SET AUTHREC** e sua sintaxe, consulte SET AUTHREC.

Para obter uma definição completa do comando PCF **MQCMD_SET_AUTH_REC** e sua sintaxe, consulte Configurar registro de autoridade.

O gerenciador de filas deve estar em execução para usar esse comando. Quando você mudou o acesso de um diretor, as mudanças são refletidas imediatamente pelo OAM.

Para fornecer aos usuários acesso a um objeto, é necessário especificar:

- O nome do gerenciador de filas que possui os objetos com os quais você está trabalhando; se você não especificar o nome de um gerenciador de filas, o gerenciador de filas padrão será assumido.
- O nome e o tipo do objeto (para identificar o objeto exclusivamente). Você especifica o nome como um *perfil*; esse é o nome explícito do objeto ou um nome genérico, incluindo caracteres curinga. Para

obter uma descrição detalhada de perfis genéricos e o uso de caracteres curinga dentro deles, consulte “Usando perfis genéricos do OAM no AIX, Linux, and Windows” na página 362.

- Um ou mais nomes de principais e de grupos aos quais a autoridade se aplica.

  Se um ID do usuário contiver espaços, coloque-o entre aspas ao usar esse comando. Em sistemas Windows, é possível qualificar um ID do usuário com um nome de domínio. Se o ID do usuário real contiver um símbolo de arroba (@), substitua-o por @@ para mostrar que ele faz parte do ID do usuário, não do delimitador entre o ID do usuário e o nome do domínio.
- Uma lista de autorizações. Cada item da lista especifica um tipo de acesso que deve ser concedido a esse objeto (ou revogado dele). Cada autorização na lista é especificada como uma palavra-chave, prefixada com um sinal de mais (+) ou um sinal de menos (-). Use um sinal de mais para incluir a autorização especificada, e um sinal de menos para remover a autorização. Não deve haver espaços entre os sinais de + ou - e a palavra-chave.

  É possível especificar qualquer número de autorizações em um único comando. Por exemplo, a lista de autorizações para permitir que um usuário ou um grupo coloque mensagens em uma fila e navegue por elas, mas revogue o acesso para obter mensagens é:

```
+browse -get +put
```

Exemplos de como usar o comando `setmqaut`

Os seguintes exemplos mostram como usar o comando `setmqaut` para conceder e revogar permissões de uso de um objeto:

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

Neste exemplo:

- `saturn.queue.manager` é o nome do gerenciador de filas
- `queue` é o tipo de objeto
- `RED.LOCAL.QUEUE` é o nome do objeto
- `groupa` é o identificador do grupo com autorizações que devem ser mudadas
- `+browse -get +put` é a lista de autorização para a fila especificada
  - `+browse` inclui autorização para navegar pelas mensagens na fila (para emitir **MQGET** com a opção de navegação)
  - `-get` remove a autorização para obter mensagens (**MQGET**) da fila
  - `+put` inclui autorização para colocar mensagens (**MQPUT**) na fila

O comando a seguir revoga a autoridade put na fila MyQueue do fvuser do diretor e dos grupos groupa e groupb. Em sistemas AIX and Linux, este comando também revoga a autoridade put para todos os principais que estão no mesmo grupo primário que fvuser.

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

Usando o comando `setmqaut` com um serviço de autorização diferente

Se você estiver usando seu próprio serviço de autorização em vez do OAM, é possível especificar o nome desse serviço no comando **setmqaut** para direcionar o comando para esse serviço. Você deverá especificar esse parâmetro se tiver vários componentes instaláveis em execução ao mesmo tempo; caso contrário, a atualização será feita no primeiro componente instalável do serviço de autorização. Por padrão, esse é o OAM fornecido.

Observações de uso para SET AUTHREC

A lista de autorizações para incluir e a lista de autorizações para remover não devem se sobrepor. Por exemplo, você não pode incluir autoridade de exibição e remover a autoridade de exibição com o mesmo comando. Essa regra se aplica mesmo que as autoridades sejam expressas usando opções diferentes. Por exemplo, o comando falhará porque a autoridade DSP se sobrepõe com autoridade a ALLADM:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

A exceção para esse comportamento de sobreposição é com a autoridade ALL. O comando a seguir inclui autoridades ALL primeiro, em seguida, remove a autoridade SETID:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

O seguinte comando remove autoridades ALL primeiro, em seguida, inclui a autoridade DSP:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

Independentemente da ordem em que são fornecidos no comando, ALL são processados primeiro.

ALW Usando perfis genéricos do OAM no AIX, Linux, and Windows

Use perfis genéricos do OAM para configurar, em uma única operação, os privilégios de um usuário para muitos objetos; em vez de ter que emitir comandos **setmqaut** separados ou comandos **SET AUTHREC** com relação a cada objeto individual quando ele for criado. Observe que no IBM MQ Appliance é possível usar apenas o comando **SET AUTHREC** .

O uso de perfis genéricos nos comandos setmqaut ou SET AUTHREC permite configurar uma autoridade genérica para todos os objetos que se ajustarem a esse perfil.

Esta coleção de tópicos descreve o uso de perfis genéricos em mais detalhes.

Usando Caracteres Curinga em Perfis OAM

O que torna um perfil genérico é o uso de caracteres especiais (caracteres curinga) no nome do perfil. Por exemplo, o caractere curinga de ponto de interrogação (?) corresponde a qualquer caractere único em um nome. Portanto, se você especificar ABC.?EF, a autorização fornecida a esse perfil se aplicará a qualquer objeto com os nomes ABC.DEF, ABC.CEF, ABC.BEF, etc.

Os caracteres curinga disponíveis são:

?

Use o ponto de interrogação (?) em vez de qualquer caractere. Por exemplo, AB.?D aplica-se aos objetos AB.CD, AB.ED e AB.FD.

*****

Use o asterisco (*) como:

- Um *qualificador* em um nome de perfil para corresponder a qualquer qualificador em um nome de objeto. Um qualificador é a parte de um nome de objeto delimitado por um ponto. Por exemplo, em ABC.DEF.GHI, os qualificadores são ABC, DEF e GHI.

  Por exemplo, ABC.*.JKL aplica-se aos objetos ABC.DEF.JKL e ABC.GHI.JKL. (Observe que ele **não** se aplica ao ABC.JKL; * usado nesse contexto sempre indica um qualificador.)
- Um caractere dentro de um qualificador em um nome de perfil para corresponder a zero ou mais caracteres dentro do qualificador em um nome de objeto.

  Por exemplo, ABC.DE*.JKL aplica-se aos objetos ABC.DE.JKL, ABC.DEF.JKL e ABC.DEGH.JKL.

******

Use o asterisco duplo (**) **uma vez** em um nome de perfil como:

- O nome do perfil inteiro para corresponder a todos os nomes de objeto. Por exemplo, se você usar `-t prcs` para identificar processos, em seguida, usar ** como o nome do perfil, você mudará as autorizações para todos os processos.
- Como o qualificador inicial, do meio ou final em um nome de perfil para corresponder a zero ou mais qualificadores em um nome de objeto. Por exemplo, `**.ABC` identifica todos os objetos com o qualificador final ABC.

É possível usar apenas o asterisco duplo ** como um qualificador completo:

```
**.DEF
  ABC.**
  A*.**
```

mas não como

```
A**
```

caso contrário, você receberá a mensagem AMQ7226E: `O nome do perfil é inválido.`

Nota: Ao usar caracteres curinga em sistemas do AIX and Linux, você **deve** colocar o nome do perfil entre aspas simples.

Prioridades do Perfil

Um ponto importante a ser entendido ao usar perfis genéricos é a prioridade que os perfis recebem ao decidirem quais autoridades aplicar a um objeto que está sendo criado. Por exemplo, suponha que você emitiu estes comandos:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

O primeiro fornece autoridade put para todas as filas para o fred principal com nomes que correspondem ao perfil AB. *; O segundo fornece autoridade get para os mesmos tipos de fila que correspondem ao perfil AB.C*.

Suponha que agora você crie uma fila chamada AB.CD. De acordo com as regras para correspondência de curinga, setmqaut poderia ser aplicado a essa fila. Portanto, ele tem autoridade put ou get?

Para localizar a resposta, você aplica a regra que, sempre que vários perfis puderem se aplicar a um objeto, **apenas o mais específico será aplicado**. A maneira de aplicação dessa regra é comparando-se os nomes dos perfis da esquerda para a direita. Sempre que forem diferentes, um caractere não genérico será mais específico do que um genérico. Portanto, neste exemplo, o AB.CD da fila possui autoridade **get** (AB.C* é mais específico do que AB.*).

Ao comparar os caracteres genéricos, a ordem de *especificidade* é:

1. ?
2. *
3. **

Fazendo Dump de Configurações do Perfil

Para obter uma definição completa do comando de controle **dmpmqaut** e sua sintaxe, consulte dmpmqaut.

Para obter uma definição completa do comando MQSC **DISPLAY AUTHREC** e sua sintaxe, consulte DISPLAY AUTHREC.

Para obter uma definição completa do comando PCF **MQCMD_INQUIRE_AUTH_RECS** e sua sintaxe, consulte Consultar registros de autoridade.

Os seguintes exemplos mostram o uso do comando de controle **dmpmqaut** para fazer dump de registros de autoridade de perfis genéricos:

1. Este exemplo faz dump de todos os registros de autoridade com um perfil que corresponde à fila a.b.c do principal user1.

```
dmpmqaut -m qm1 -n a.b.c -t q -p user1
```

   O dump resultante é semelhante a isto:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

   Nota: Embora os usuários no AIX and Linux possam usar a opção `-p` para o comando **dmpmqaut**, eles devem usar `-g groupname` no lugar ao definir autorizações.

2. Este exemplo faz dump de todos os registros de autoridade com um perfil que corresponde à fila a.b.c.

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

   O dump resultante é semelhante a isto:

```
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. Este exemplo faz dump de todos os registros de autoridade para o perfil a.b. *, de fila do tipo

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

   O dump resultante é semelhante a isto:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. Este exemplo faz dump de todos os registros de autoridade para o gerenciador de filas qmX.

```
dmpmqaut -m qmX
```

   O dump resultante é semelhante a isto:

```
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. Este exemplo faz dump de todos os nomes de perfis e tipos de objetos para o gerenciador de filas qmX.

   ```
   dmpmqaut -m qmX -l
   ```

   O dump resultante é semelhante a isto:

   ```
   profile: q1, type: queue
   profile: q*, type: queue
   profile: name.*, type: namelist
   profile: pr1, type: process
   ```

Nota: Somente para o IBM MQ for Windows, todos os principais exibidos incluem informações de domínio, por exemplo:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW *Usando caracteres curinga em perfis do OAM no AIX, Linux, and Windows*

Use caracteres curinga em um nome de perfil do Object Authority Manager (OAM) para que esse perfil seja aplicável a mais de um objeto.

O que torna um perfil genérico é o uso de caracteres especiais (caracteres curinga) no nome do perfil. Por exemplo, o caractere curinga de ponto de interrogação (?) corresponde a qualquer caractere único em um nome. Portanto, se você especificar `ABC.?EF`, a autorização fornecida a esse perfil se aplicará a qualquer objeto com os nomes `ABC.DEF`, `ABC.CEF`, `ABC.BEF`, etc.

Os caracteres curinga disponíveis são:

?

Use o ponto de interrogação (?) em vez de qualquer caractere. Por exemplo, `AB.?D` aplica-se aos objetos `AB.CD`, `AB.ED` e `AB.FD`.

*****

Use o asterisco (*) como:

- Um *qualificador* em um nome de perfil para corresponder a qualquer qualificador em um nome de objeto. Um qualificador é a parte de um nome de objeto delimitado por um ponto. Por exemplo, em `ABC.DEF.GHI`, os qualificadores são `ABC`, `DEF` e `GHI`.

  Por exemplo, `ABC.*.JKL` aplica-se aos objetos `ABC.DEF.JKL` e `ABC.GHI.JKL`. (Observe que ele **não** se aplica ao `ABC.JKL`; * usado nesse contexto sempre indica um qualificador.)
- Um caractere dentro de um qualificador em um nome de perfil para corresponder a zero ou mais caracteres dentro do qualificador em um nome de objeto.

  Por exemplo, `ABC.DE*.JKL` aplica-se aos objetos `ABC.DE.JKL`, `ABC.DEF.JKL` e `ABC.DEGH.JKL`.

**

Use o asterisco duplo (**) **uma vez** em um nome de perfil como:

- O nome do perfil inteiro para corresponder a todos os nomes de objeto. Por exemplo, se você usar `-t prcs` para identificar processos, em seguida, usar ** como o nome do perfil, você mudará as autorizações para todos os processos.
- Como o qualificador inicial, do meio ou final em um nome de perfil para corresponder a zero ou mais qualificadores em um nome de objeto. Por exemplo, `**.ABC` identifica todos os objetos com o qualificador final ABC.

Nota: Ao usar caracteres curinga em sistemas do AIX and Linux, você **deve** colocar o nome do perfil entre aspas simples.

ALW *Prioridades de perfil no AIX, Linux, and Windows*

Mais de um perfil genérico pode aplicar-se a um único objeto. Quando for esse o caso, a regra mais específica se aplicará.

Um ponto importante a ser entendido ao usar perfis genéricos é a prioridade que os perfis recebem ao decidirem quais autoridades aplicar a um objeto que está sendo criado. Por exemplo, suponha que você emitiu estes comandos:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

O primeiro fornece autoridade put para todas as filas para o fred principal com nomes que correspondem ao perfil AB. *; O segundo fornece autoridade get para os mesmos tipos de fila que correspondem ao perfil AB.C*.

Suponha que agora você crie uma fila chamada AB.CD. De acordo com as regras para correspondência de curinga, setmqaut poderia ser aplicado a essa fila. Portanto, ele tem autoridade put ou get?

Para localizar a resposta, você aplica a regra que, sempre que vários perfis puderem se aplicar a um objeto, **apenas o mais específico será aplicado**. A maneira de aplicação dessa regra é comparando-se os nomes dos perfis da esquerda para a direita. Sempre que forem diferentes, um caractere não genérico será mais específico do que um genérico. Portanto, neste exemplo, o AB.CD da fila possui autoridade **get** (AB.C* é mais específico do que AB.*).

Ao comparar os caracteres genéricos, a ordem de *especificidade* é:

1. ?
2. *
3. **

Consulte SET AUTHREC para as informações equivalentes ao usar este comando MQSC.

ALW *Fazendo dump das configurações de perfil no AIX, Linux, and Windows*

Use o comando de controle **dmpmqaut** , o comando MQSC **DISPLAY AUTHREC** ou o comando PCF **MQCMD_INQUIRE_AUTH_RECS** para fazer dump das autorizações atuais associadas a um perfil especificado.. Observe que no IBM MQ Appliance é possível usar apenas o comando **DISPLAY AUTHREC** .

Para obter uma definição completa do comando de controle **dmpmqaut** e sua sintaxe, consulte dmpmqaut.

Para obter uma definição completa do comando MQSC **DISPLAY AUTHREC** e sua sintaxe, consulte DISPLAY AUTHREC.

Para obter uma definição completa do comando PCF **MQCMD_INQUIRE_AUTH_RECS** e sua sintaxe, consulte Consultar registros de autoridade.

Os seguintes exemplos mostram o uso do comando de controle **dmpmqaut** para fazer dump de registros de autoridade de perfis genéricos:

1. Este exemplo faz dump de todos os registros de autoridade com um perfil que corresponde à fila a.b.c do principal user1.

```
dmpmqaut -m qm1 -n a.b.c -t q -p user1
```

   O dump resultante é semelhante a este exemplo:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

   Nota: Usuários do AIX and Linux não podem usar a opção `-p`; eles devem usar `-g groupname`.

2. Este exemplo faz dump de todos os registros de autoridade com um perfil que corresponde à fila a.b.c.

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

   O dump resultante é semelhante a este exemplo:

```
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. Este exemplo faz dump de todos os registros de autoridade para o perfil a.b. *, de fila do tipo

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

   O dump resultante é semelhante a este exemplo:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. Este exemplo faz dump de todos os registros de autoridade para o gerenciador de filas qmX.

```
dmpmqaut -m qmX
```

   O dump resultante é semelhante a este exemplo:

```
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. Este exemplo faz dump de todos os nomes de perfis e tipos de objetos para o gerenciador de filas qmX.

   ```
   dmpmqaut -m qmX -l
   ```

   O dump resultante é semelhante a este exemplo:

   ```
   profile: q1, type: queue
   profile: q*, type: queue
   profile: name.*, type: namelist
   profile: pr1, type: process
   ```

Nota: Somente para o IBM MQ for Windows, todos os principais exibidos incluem informações de domínio, por exemplo:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW Exibindo configurações de acesso no AIX, Linux, and Windows

Use o comando de controle **dspmqaut** , o comando MQSC **DISPLAY AUTHREC** ou o comando PCF **MQCMD_INQUIRE_ENTITY_AUTH** para visualizar as autorizações que um proprietário ou grupo específico possui para um objeto específico Observe que no IBM MQ Appliance, é possível usar somente o comando **DISPLAY AUTHREC**.

O gerenciador de filas deve estar em execução para usar esse comando. Ao mudar o acesso de um diretor, as mudanças são refletidas imediatamente pelo OAM. A autorização pode ser exibida para apenas um grupo ou diretor de cada vez.

Para obter uma definição completa do comando de controle **dmpmqaut** e sua sintaxe, consulte dmpmqaut.

Para obter uma definição completa do comando MQSC **DISPLAY AUTHREC** e sua sintaxe, consulte DISPLAY AUTHREC.

Para obter uma definição completa do comando PCF **MQCMD_INQUIRE_AUTH_RECS** e sua sintaxe, consulte Consultar registros de autoridade.

O exemplo a seguir mostra o uso do comando de controle **dspmqaut** para exibir as autorizações que o grupo `GpAdmin` tem para uma definição de processo chamada `Annuities` que está no gerenciador de filas `QueueMan1`

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

ALW Mudando e revogando o acesso a um objeto do IBM MQ no AIX, Linux, and Windows

Para mudar o nível de acesso que um usuário ou grupo tem para um objeto, use o comando de controle **setmqaut**, o comando MQSC **DELETE AUTHREC** ou o comando PCF **MQCMD_DELETE_AUTH_REC**.

MQ Appliance Observe que no IBM MQ Appliance é possível usar apenas o comando **DELETE AUTHREC** .

O processo de remover o usuário de um grupo é descrito em:

- Windows "Criando e gerenciando grupos no Windows" na página 153
- AIX "Criando e gerenciando grupos no AIX" na página 152
- Linux "Criando e gerenciando grupos no Linux" na página 153

.

O ID do usuário que cria um objeto do IBM MQ é concedido autoridades de controle totais para esse objeto. Se você remover este ID do usuário do grupo mqm local (ou o grupo Administradores em sistemas Windows) essas autoridades não serão revogadas. Use o comando de controle **setmqaut** ou o comando PCF **MQCMD_DELETE_AUTH_REC** para revogar o acesso a um objeto para o ID do usuário que o criou, depois de removê-lo do grupo mqm ou do grupo de Administradores.

Para obter uma definição completa do comando de controle setmqaut e sua sintaxe, consulte setmqaut.

Para obter uma definição completa do comando MQSC **DELETE AUTHREC** e sua sintaxe, consulte DELETE AUTHREC.

Para obter uma definição completa do comando PCF **MQCMD_DELETE_AUTH_REC** e sua sintaxe, consulte Excluir registro de autoridade.

Windows No Windows, a partir do IBM MQ 8.0, é possível excluir as entradas OAM correspondentes a uma conta do usuário específica do Windows a qualquer momento usando o parâmetro **-u** *SID* do **setmqaut**.

Antes do IBM MQ 8.0, você teve de excluir as entradas OAM correspondentes a uma conta de usuário específica do Windows antes de excluir o perfil do usuário. Foi impossível remover as entradas OAM após remover a conta do usuário.

ALW Evitando verificações de acesso de segurança nos sistemas AIX, Linux, and Windows

Nota: Este tópico descreve a funcionalidade que não é recomendada para ser ativada Para desativar a verificação de segurança, é possível desativar o OAM (Object Authority Manager). Isso pode ser adequado para um ambiente de teste. Quando desativado, o gerenciador de filas não poderá mais executar verificações de autorização ou de autenticação de conexão. TLS, registros de Autenticação de Canal e saídas de segurança ainda podem ser usados. Depois de desativar ou remover o OAM, não é possível incluir um OAM em um gerenciador de filas existente.

Se você decidir que não deseja executar verificações de segurança (por exemplo, em um ambiente de teste), poderá desativar o OAM em uma de duas maneiras:

- Antes de criar um gerenciador de filas, configure a variável de ambiente do sistema operacional **MQSNOAUT**.

  Para obter informações sobre as implicações de configurar a variável de ambiente **MQSNOAUT** e como configurar **MQSNOAUT** em AIX, Linux, and Windows, consulte Descrições de variáveis de ambiente.
- Edite o arquivo de configuração do gerenciador de filas para remover o serviço.

Aviso: Quando um OAM é removido, ele não pode ser colocado de volta em um gerenciador de filas existente. Isso ocorre porque o OAM precisa estar ativo no horário de criação do objeto. Para usar o IBM MQ OAM novamente depois que ele foi removido, reconstrua o gerenciador de filas.

Se você usar o comando **setmqaut** ou **dspmqaut** enquanto o OAM estiver desativado, observe os pontos a seguir:

- O OAM não valida o diretor ou o grupo especificado, isto é, o comando pode aceitar valores inválidos.
- O OAM não executa verificações de segurança e indica que todos os principais e grupos têm autorização para executar todas as operações de objeto aplicáveis.
- Quaisquer credenciais transmitidas para o OAM para verificações de autenticação não são validadas.

Conceitos relacionados

Serviços e componentes instaláveis para o AIX, Linux, and Windows

Tarefas relacionadas

Configurando serviços instaláveis

Referências relacionadas

Informações de referência de serviços instaláveis

Concedendo acesso necessário para recursos

Use este tópico para determinar quais tarefas executar para aplicar a segurança para o seu sistema IBM MQ.

Sobre esta tarefa

Durante esta tarefa, você decide quais ações são necessárias para aplicar o nível apropriado de segurança para os elementos de sua instalação do IBM MQ. Cada tarefa individual para a qual você é encaminhado fornece instruções passo a passo para todas as plataformas.

Procedimento

1. Você precisa limitar o acesso ao gerenciador de filas para determinados usuários?
   a) Não: não executar ação adicional.
   b) Sim: Acesse a próxima pergunta.
2. Esses usuários precisam de acesso administrativo parcial em um subconjunto de recursos de gerenciadores de filas?
   a) Não: Acesse a próxima pergunta.
   b) Sim: Consulte “Concedendo Acesso Administrativo Parcial em um Subconjunto de Recursos do Gerenciador de Filas” na página 371.
3. Esses usuários precisam de acesso administrativo total em um subconjunto de recursos de gerenciadores de filas?
   a) Não: Acesse a próxima pergunta.
   b) Sim: Consulte “Concedendo Acesso Administrativo Total em um Subconjunto de Recursos do Gerenciador de Filas” na página 379.
4. Esses usuários precisam de acesso somente leitura a todos os recursos de gerenciadores de filas?
   a) Não: Acesse a próxima pergunta.
   b) Sim: Consulte “Concedendo Acesso somente Leitura a todos os Recursos em um Gerenciador de Filas” na página 385.
5. Esses usuários precisam de acesso administrativo total em todos recursos de gerenciadores de filas?
   a) Não: Acesse a próxima pergunta.
   b) Sim: Consulte “Concedendo Acesso Administrativo Total a todos os Recursos em um Gerenciador de Filas” na página 387.
6. Você precisa que os aplicativos de usuários se conectem ao gerenciador de filas?
   a) Não: Desative a conectividade, conforme descrito em “Removendo a Conectividade com o Gerenciador de Filas” na página 388

b) Sim: Consulte “Permitindo que Aplicativos de Usuário se Conectem ao Gerenciador de Filas” na página 389.

Multi z/OS Concedendo Acesso Administrativo Parcial em um Subconjunto de Recursos do Gerenciador de Filas

É necessário fornecer a determinados usuários acesso administrativo parcial a alguns recursos do gerenciador de filas, mas não a todos. Use esta tabela para determinar as ações que precisam ser executadas.

Tabela 72. Concedendo acesso administrativo parcial para um subconjunto de recursos do gerenciador de filas

Os usuários precisam administrar objetos deste tipo	**Executar esta ação**
Filas	Conceda acesso administrativo parcial às filas necessárias, conforme descrito em “Concedendo Acesso Administrativo Limitado a algumas Filas” na página 371
tópicos	Conceda acesso administrativo parcial aos tópicos necessários, conforme descrito em “Concedendo Acesso Administrativo Limitado a alguns Tópicos” na página 373
Canais	Conceda acesso administrativo parcial aos canais necessários, conforme descrito em “Concedendo Acesso Administrativo Limitado a alguns Canais” na página 374
O gerenciador de filas	Conceda acesso administrativo parcial ao gerenciador de filas, conforme descrito em “Concedendo Acesso Administrativo Limitado a um Gerenciador de Filas” na página 375
Processos	Conceda acesso administrativo parcial aos processos necessários, conforme descrito em “Concedendo Acesso Administrativo Limitado a alguns Processos” na página 376
Listas de Nomes	Conceda acesso administrativo parcial às listas de nomes necessárias, conforme descrito em “Concedendo Acesso Administrativo Limitado a algumas Listas de Nomes” na página 377
Serviços	Conceda acesso administrativo parcial aos serviços necessários, conforme descrito em “Concedendo Acesso Administrativo Limitado a alguns Serviços” na página 378

Concedendo Acesso Administrativo Limitado a algumas Filas

Conceda acesso administrativo parcial a algumas filas em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo limitado a algumas filas para algumas ações, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Nota: MQ Appliance No IBM MQ Appliance é possível usar apenas o comando **SET AUTHREC**.

Procedimento

- ALW
  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
  ```

- IBM i
  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Para o z/OS, emita os comandos a seguir para conceder acesso a uma fila especificada:

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Para especificar quais comandos MQSC o usuário pode executar na fila, emita os comandos a seguir para cada comando MQSC:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction. QType UACC(NONE)
  PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Para permitir que o usuário use o comando DISPLAY QUEUE, emita os seguintes comandos:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY. QType UACC(NONE)
  PERMIT QMgrName.DISPLAY. QType CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

  ReqdAction
  A ação que você está permitindo que o grupo execute:

  – ALW Nos sistemas AIX, Linux, and Windows, qualquer combinação das autorizações a seguir: +chg, +clr, +dlt, +dsp. A autorização +alladm é equivalente a +chg +clr +dlt +dsp.

  – IBM i No IBM i, qualquer combinação das autorizações a seguir: *ADMCHG, *ADMCLR, *ADMDLT, *ADMDSP. A autorização *ALLADM é equivalente a todas essas autorizações individuais.

  – z/OS No z/OS, um dos valores ALTER, CLEAR, DELETE ou MOVE.

  Nota: Conceder +crt para filas torna indiretamente o usuário ou o grupo um administrador. Não use a autoridade +crt para conceder acesso administrativo limitado a algumas filas.

QType

Para o comando DISPLAY, um dos valores QUEUE, QLOCAL, QALIAS, QMODEL, QREMOTE ou QCLUSTER.

Para outros valores de *ReqdAction*, um dos valores QLOCAL, QALIAS, QMODEL ou QREMOTE.

Concedendo Acesso Administrativo Limitado a alguns Tópicos

Conceda acesso administrativo parcial a alguns tópicos em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo limitado a alguns tópicos para algumas ações, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW

  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
  ```

- IBM i

  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Para z/OS, emita os seguintes comandos:

  ```
  RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
  PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Esses comandos concedem acesso ao tópico especificado. Para determinar quais comandos MQSC o usuário pode executar no tópico, emita os seguintes comandos para cada comando MQSC:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.TOPIC UACC(NONE)
  PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Para permitir que o usuário use o comando DISPLAY TOPIC, emita os seguintes comandos:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.TOPIC UACC(NONE)
  PERMIT QMgrName.DISPLAY.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

ReqdAction
A ação que você está permitindo que o grupo execute:

- ALW Em sistemas AIX, Linux, and Windows, qualquer combinação das autorizações a seguir: +chg, +clr, +crt, +dlt, +dsp. +ctrl. A autorização +alladm é equivalente a +chg +clr +dlt +dsp.
- IBM i No IBM i, qualquer combinação das autorizações a seguir: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL. A autorização *ALLADM é equivalente a todas essas autorizações individuais.
- z/OS No z/OS, um dos valores ALTER, CLEAR, DEFINE, DELETE ou MOVE.

Concedendo Acesso Administrativo Limitado a alguns Canais

Conceda acesso administrativo parcial a alguns canais em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo limitado a alguns canais para algumas ações, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- IBM i
  No IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS No z/OS:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Esses comandos concedem acesso ao canal especificado. Para determinar quais comandos MQSC o usuário pode executar no canal, emita os seguintes comandos para cada comando MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.CHANNEL UACC(NONE)
PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Para permitir que o usuário use o comando DISPLAY CHANNEL, emita os seguintes comandos:

```
RDEFINE MQCMDS QMgrName.DISPLAY.CHANNEL UACC(NONE)
PERMIT QMgrName.DISPLAY.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

ReqdAction
A ação que você está permitindo que o grupo execute:

- ALW No AIX, Linux, and Windows, qualquer combinação das autorizações a seguir: +chg, +clr, +crt, +dlt, +dsp. +ctrl, +ctrlx. A autorização +alladm é equivalente a +chg +clr +dlt +dsp.
- IBM i No IBM i, qualquer combinação das autorizações a seguir: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. A autorização *ALLADM é equivalente a todas essas autorizações individuais.
- z/OS No z/OS, um dos valores ALTER, CLEAR, DEFINE, DELETE ou MOVE.

Concedendo Acesso Administrativo Limitado a um Gerenciador de Filas

Conceda acesso administrativo parcial a um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo limitado para executar algumas ações no gerenciador de filas, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
```

- IBM i
  No IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS
  No z/OS:

  Para determinar quais comandos MQSC você pode executar no gerenciador de filas, emita os comandos a seguir para cada comando MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.QMGR UACC(NONE)
PERMIT QMgrName. ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Para permitir que o usuário use o comando DISPLAY QMGR, emita os seguintes comandos:

```
RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

ReqdAction
A ação que você está permitindo que o grupo execute:

- ALW No AIX, Linux, and Windows, qualquer combinação das autorizações a seguir: +chg, +clr, +crt, +dlt, +dsp. A autorização +alladm é equivalente a +chg +clr +dlt +dsp.

  Embora +set seja uma autorização do MQI e não considerada administrativa normalmente, conceder +set no gerenciador de filas pode levar indiretamente à autoridade administrativa total. Não conceda +set a usuários e aplicativos ordinários.
- IBM i No IBM i, qualquer combinação das autorizações a seguir: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. A autorização *ALLADM é equivalente a todas essas autorizações individuais.

Concedendo Acesso Administrativo Limitado a alguns Processos

Conceda acesso administrativo parcial a alguns processos em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo limitado a alguns processos para algumas ações, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
```

- IBM i
  No IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS No z/OS:

```
RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Esses comandos concedem acesso ao canal especificado. Para determinar quais comandos MQSC o usuário pode executar no canal, emita os seguintes comandos para cada comando MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.PROCESS UACC(NONE)
PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Para permitir que o usuário use o comando DISPLAY PROCESS, emita os seguintes comandos:

```
RDEFINE MQCMDS QMgrName.DISPLAY.PROCESS UACC(NONE)
PERMIT QMgrName.DISPLAY.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas.

z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

ReqdAction
A ação que você está permitindo que o grupo execute:

- ALW No AIX, Linux, and Windows, qualquer combinação das autorizações a seguir: +chg, +clr, +crt, +dlt, +dsp. A autorização +alladm é equivalente a +chg +clr +dlt +dsp.
- IBM i No IBM i, qualquer combinação das autorizações a seguir: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. A autorização *ALLADM é equivalente a todas essas autorizações individuais.
- z/OS No z/OS, um dos valores ALTER, CLEAR, DEFINE, DELETE ou MOVE.

Concedendo Acesso Administrativo Limitado a algumas Listas de Nomes

Conceda acesso administrativo parcial a algumas listas de nomes em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo limitado a algumas listas de nomes para algumas ações, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
  ```

- IBM i
  No IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS No z/OS:

  ```
  RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
  PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

Esses comandos concedem acesso à lista de nomes especificada. Para determinar quais comandos MQSC o usuário pode executar na lista de nomes, emita os seguintes comandos para cada comando MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.NAMELIST UACC(NONE)
PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Para permitir que o usuário use o comando DISPLAY NAMELIST, emita os seguintes comandos:

```
RDEFINE MQCMDS QMgrName.DISPLAY.NAMELIST UACC(NONE)
PERMIT QMgrName.DISPLAY.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas.

z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

ReqdAction
A ação que você está permitindo que o grupo execute:

- ALW No AIX, Linux, and Windows, qualquer combinação das autorizações a seguir: +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp. A autorização +alladm é equivalente a +chg +clr +dlt +dsp.
- IBM i No IBM i, qualquer combinação das autorizações a seguir: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. A autorização *ALLADM é equivalente a todas essas autorizações individuais.
- z/OS No z/OS, um dos valores ALTER, CLEAR, DEFINE, DELETE ou MOVE.

Concedendo Acesso Administrativo Limitado a alguns Serviços

Conceda acesso administrativo parcial a alguns serviços em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo limitado a alguns serviços para algumas ações, use os comandos apropriados de seu sistema operacional. z/OS Note que os objetos de serviço não existem no z/OS.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- No IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS No z/OS:

  Esses comandos concedem acesso ao serviço especificado. Para determinar quais comandos MQSC o usuário pode executar no serviço, emita os seguintes comandos para cada comando MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.SERVICE UACC(NONE)
PERMIT QMgrName. ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Para permitir que o usuário use o comando DISPLAY SERVICE, emita os seguintes comandos:

```
RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

  ReqdAction
  A ação que você está permitindo que o grupo execute:

  - ALW Nos sistemas AIX, Linux, and Windows, qualquer combinação das autorizações a seguir: +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp. A autorização +alladm é equivalente a +chg +clr +dlt +dsp.
  - IBM i No IBM i, qualquer combinação das autorizações a seguir: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. A autorização *ALLADM é equivalente a todas essas autorizações individuais.

Concedendo Acesso Administrativo Total em um Subconjunto de Recursos do Gerenciador de Filas

É necessário fornecer a determinados usuários acesso administrativo total a alguns recursos do gerenciador de filas, mas não a todos. Use estas tabelas para determinar as ações que precisam ser executadas.

Tabela 73. Concedendo acesso administrativo completo para um subconjunto de recursos do gerenciador de filas

Os usuários precisam administrar objetos deste tipo	**Executar esta ação**
Filas	Conceda acesso administrativo total às filas necessárias, conforme descrito em “Concedendo Acesso Administrativo Total a algumas Filas” na página 380
tópicos	Conceda acesso administrativo total aos tópicos necessários, conforme descrito em “Concedendo Acesso Administrativo Total a alguns Tópicos” na página 381

Tabela 73. Concedendo acesso administrativo completo para um subconjunto de recursos do gerenciador de filas (continuação)

Os usuários precisam administrar objetos deste tipo	**Executar esta ação**
Canais	Conceda acesso administrativo total aos canais necessários, conforme descrito em "Concedendo Acesso Administrativo Total a alguns Canais" na página 382
O gerenciador de filas	Conceda acesso administrativo total ao gerenciador de filas, conforme descrito em "Concedendo Acesso Administrativo Total a um Gerenciador de Filas" na página 382
Processos	Conceda acesso administrativo total aos processos necessários, conforme descrito em "Concedendo Acesso Administrativo Total a alguns Processos" na página 383
Listas de Nomes	Conceda acesso administrativo total às listas de nomes necessárias, conforme descrito em "Concedendo Acesso Administrativo Total a algumas Listas de Nomes" na página 384
Serviços	Conceda acesso administrativo total aos serviços necessários, conforme descrito em "Concedendo Acesso Administrativo Total a alguns Serviços" na página 384

Concedendo Acesso Administrativo Total a algumas Filas

Conceda acesso administrativo total a algumas filas em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo total a algumas filas, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
```

- IBM i
  No IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrName
')
```

- z/OS
  No z/OS:

```
RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas.

z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

Concedendo Acesso Administrativo Total a alguns Tópicos

Conceda acesso administrativo total a alguns tópicos em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo total a alguns tópicos para algumas ações, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
  ```

- IBM i
  No IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  No z/OS:

  ```
  RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
  PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo Acesso Administrativo Total a alguns Canais

Conceda acesso administrativo total a alguns canais em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo total a alguns canais, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
  ```

- IBM i
  No IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  No z/OS:

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo Acesso Administrativo Total a um Gerenciador de Filas

Conceda acesso administrativo total a um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo total ao gerenciador de filas, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

```
setmqaut -m QMgrName -t qmgr -g GroupName +alladm
```

- IBM i

  No IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  No z/OS:

  ```
  RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
  PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo Acesso Administrativo Total a alguns Processos

Conceda acesso administrativo total a alguns processos em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo total a alguns processos, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW

  No AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
  ```

- IBM i

  No IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  No z/OS:

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas.

z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

Concedendo Acesso Administrativo Total a algumas Listas de Nomes

Conceda acesso administrativo total a algumas listas de nomes em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo total a algumas listas de nomes, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  No AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
```

- IBM i
  No IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  No z/OS:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo Acesso Administrativo Total a alguns Serviços

Conceda acesso administrativo total a alguns serviços em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder acesso administrativo total a alguns serviços, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW

  No AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
  ```

- IBM i

  No IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  No z/OS:

  ```
  RDEFINE MQADMIN QMgrName.SERVICE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.SERVICE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo Acesso somente Leitura a todos os Recursos em um Gerenciador de Filas

Conceda acesso somente leitura a todos os recursos em um gerenciador de filas, a cada usuário ou grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Use o assistente Incluir autoridades baseadas na função ou os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Depois de mudar quaisquer detalhes de autorização, execute uma atualização de segurança usando o comando REFRESH SECURITY.

Procedimento

- Usando o assistente:

a) Na área de janela do Navegador IBM MQ Explorer , clique com o botão direito do mouse no gerenciador de filas e clique em **Autoridades de Objetos** > **Incluir Autoridades Baseadas em Função**

   O assistente Incluir Autoridades Baseadas na Função é aberto.

- ALW

  Para sistemas AIX, Linux, and Windows, emita os comandos a seguir:

```
setmqaut -m QMgrName -n ** -t queue -g GroupName +browse +dsp
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get
+put
setmqaut -m QMgrName -n ** -t topic -g GroupName +dsp
setmqaut -m QMgrName -n ** -t channel -g GroupName +dsp
setmqaut -m QMgrName -n ** -t clntconn -g GroupName +dsp
setmqaut -m QMgrName -n ** -t authinfo -g GroupName +dsp
setmqaut -m QMgrName -n ** -t listener -g GroupName +dsp
setmqaut -m QMgrName -n ** -t namelist -g GroupName +dsp
setmqaut -m QMgrName -n ** -t process -g GroupName +dsp
setmqaut -m QMgrName -n ** -t service -g GroupName +dsp
setmqaut -m QMgrName -t qmgr -g GroupName +dsp +inq +connect
```

  As autoridades específicas para o SYSTEM.ADMIN.COMMAND.QUEUE e SYSTEM.MQEXPLORER.REPLY.MODEL são necessários apenas se você desejar utilizar o IBM MQ Explorer

- IBM i

  Para IBM i, emita os seguintes comandos:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*Q) USER('GroupName') AUT(*ADMDSP *BROWSE) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*TOPIC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CHL) USER('GroupName') AUT(*ADMDSP *INQ) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*CLTCN) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*AUTHINFO) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*LSR) USER('GroupName') AUT(*ADMDSP)MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*NMLIST) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ(*ALL) OBJTYPE(*SVC) USER('GroupName') AUT(*ADMDSP) MQMNAME('QMgrName')
GRTMQMAUT OBJ('object-name') OBJTYPE(*MQM) USER('GroupName') AUT(*ADMDSP *CONNECT *INQ)
MQMNAME('QMgrName')
```

- z/OS

  Para z/OS, emita os seguintes comandos:

```
RDEFINE MQQUEUE QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQQUEUE) ID(GroupName) ACCESS(READ)
RDEFINE MXTOPIC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MXTOPIC) ID(GroupName) ACCESS(READ)
RDEFINE MQPROC QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQPROC) ID(GroupName) ACCESS(READ)
RDEFINE MQNLIST QMgrName.** UACC(NONE)
PERMIT QMgrName.** CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName

  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

GroupName
O nome do grupo que receberá acesso.

Concedendo Acesso Administrativo Total a todos os Recursos em um Gerenciador de Filas

Conceda acesso administrativo total a todos os recursos em um gerenciador de filas, a cada usuário ou grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

É possível usar o assistente Incluir autoridades baseadas em função ou os comandos apropriados para seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Notas: ALW

1. Se você estiver usando o **runmqsc** para administrar o gerenciador de fila em vez do IBM MQ Explorer, deverá conceder autoridade para consultar, obter e procurar o SYSTEM.MQSC.REPLY.QUEUE, e você não precisa conceder nenhuma autoridade no SYSTEM.MQEXPLORER.REPLY.MODEL fila.
2. Ao conceder a um usuário acesso a todos os recursos em um gerenciador de filas, há alguns comandos que o usuário não pode executar, a menos que esse usuário tenha acesso de leitura ao arquivo `qm.ini`. Isso ocorre devido a restrições sobre os usuários não `mqm` poderem ler o arquivo `qm.ini`.

   O usuário não pode emitir os comandos a seguir, a menos que você tenha concedido a esse usuário acesso de leitura ao arquivo qm.ini:

   - Definindo um canal configurado para usar TLS
   - Definindo um canal usando variáveis de inserção de configuração automática definidas no `qm.ini`

Procedimento

- Se você estiver usando o assistente, na área de janela do IBM MQ Explorer Navigator, clique com o botão direito no Gerenciador de Filas e clique em **Autoridades do objeto** > **Incluir autoridades baseadas em função**.

  O assistente Incluir Autoridades Baseadas na Função é aberto.

- Linux AIX

  Para sistemas AIX and Linux, emita os comandos a seguir:

```
setmqaut -m QMgrName -n '**' -t queue -g GroupName +alladm +browse
setmqaut -m QMgrName -n @class -t queue -g GroupName +crt
setmqaut -m QMgrName -n SYSTEM.ADMIN.COMMAND.QUEUE -t queue -g GroupName +dsp +inq +put
setmqaut -m QMgrName -n SYSTEM.MQEXPLORER.REPLY.MODEL -t queue -g GroupName +dsp +inq +get +put
setmqaut -m QMgrName -n '**' -t topic -g GroupName +alladm
setmqaut -m QMgrName -n @class -t topic -g GroupName +crt
setmqaut -m QMgrName -n '**' -t channel -g GroupName +alladm
setmqaut -m QMgrName -n @class -t channel -g GroupName +crt
setmqaut -m QMgrName -n '**' -t clntconn -g GroupName +alladm
setmqaut -m QMgrName -n @class -t clntconn -g GroupName +crt
setmqaut -m QMgrName -n '**' -t authinfo -g GroupName +alladm
setmqaut -m QMgrName -n @class -t authinfo -g GroupName +crt
setmqaut -m QMgrName -n '**' -t listener -g GroupName +alladm
setmqaut -m QMgrName -n @class -t listener -g GroupName +crt
setmqaut -m QMgrName -n '**' -t namelist -g GroupName +alladm
setmqaut -m QMgrName -n @class -t namelist -g GroupName +crt
setmqaut -m QMgrName -n '**' -t process -g GroupName +alladm
setmqaut -m QMgrName -n @class -t process -g GroupName +crt
setmqaut -m QMgrName -n '**' -t service -g GroupName +alladm
setmqaut -m QMgrName -n @class -t service -g GroupName +crt
setmqaut -m QMgrName -t qmgr -g GroupName +alladm +connect
```

  Consulte **setmqaut** para obter mais informações sobre `@class`

- Windows

Para sistemas Windows, emita os mesmos comandos para sistemas AIX and Linux, mas usando o nome do perfil @CLASS em vez de @class.

- IBM i

  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER(' GroupName ') AUT(*ALLADM) MQMNAME(' QMgrName ')
  ```

- z/OS

  Para z/OS, emita os seguintes comandos:

  ```
  RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
  PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas.

  z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  GroupName
  O nome do grupo que receberá acesso.

Removendo a Conectividade com o Gerenciador de Filas

Para que aplicativos de usuário não se conectem ao gerenciador de filas, remova sua autoridade de conexão com ele.

Sobre esta tarefa

Revogue a autoridade de todos os usuários de conexão com o gerenciador de filas usando o comando apropriado de seu sistema operacional.

No Multiplataformas, também é possível usar o comando DELETE AUTHREC.

Nota: No IBM MQ Appliance é possível usar somente o comando **DELETE AUTHREC**.

Procedimento

- ALW

  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName -connect
  ```

- IBM i

  Para o IBM i, emita o comando a seguir:

  ```
  RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
  ```

- z/OS

  Para z/OS, emita os seguintes comandos:

  ```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
  RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
  RDEFINE MQCONN QMgrName.CICS UACC(NONE)
  RDEFINE MQCONN QMgrName.IMS UACC(NONE)
  ```

  Não emita comandos PERMIT.

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas.

z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

GroupName
O nome do grupo cujo acesso será negado.

Permitindo que Aplicativos de Usuário se Conectem ao Gerenciador de Filas

Você deseja permitir que o aplicativo de usuário se conecte ao gerenciador de filas. Use as tabelas deste tópico para determinar quais ações executar.

Primeiro, determine se os aplicativos clientes se conectarão ao seu Gerenciador de Filas.

Se nenhum dos aplicativos que irão se conectar ao gerenciador de filas for aplicativo cliente, desative o acesso remoto, conforme descrito em "Desativando o Acesso Remoto ao Gerenciador de Filas" na página 396.

Se um ou mais dos aplicativos que irão se conectar ao gerenciador de filas forem aplicativos clientes, proteja a conectividade remota, conforme descrito em "Protegendo a Conectividade Remota no Gerenciador de Filas" na página 389.

Em ambos os casos, configure a segurança de conexão, conforme descrito em "Configurando a Segurança de Conexão" na página 397

Para controlar o acesso a recursos de cada usuário que se conectar ao gerenciador de filas, consulte a tabela a seguir. Se a instrução na primeira coluna for verdadeira, execute a ação listada na segunda coluna.

Declaração	Execute esta ação
Você tem aplicativos que usam filas	Consulte "Controlando o Acesso de Usuário a Filas" na página 398
Você tem aplicativos que usam tópicos	Consulte o "Controlando o Acesso de Usuário aos Tópicos" na página 403.
Você tem aplicativos que consultam o objeto de gerenciador de filas	Consulte o "Concedendo autoridade para consultar em um gerenciador de filas" na página 405.
Você tem aplicativos que usam objetos de processos	Consulte "Concedendo autoridade para acessar processos" na página 406
Você tem aplicativos que usam listas de nomes	Consulte "Concedendo autoridade para acessar listas de nomes" na página 407

Protegendo a Conectividade Remota no Gerenciador de Filas

É possível assegurar a conectividade remota com o gerenciador de filas usando TLS, uma saída de segurança, registros de autenticação de canal ou uma combinação desses métodos.

Sobre esta tarefa

Você conecta um cliente ao gerenciador de filas usando um canal de conexão do cliente na estação de trabalho do cliente e um canal de conexão do servidor no servidor. Proteja essas conexões de uma das seguintes maneiras.

Procedimento

1. Usando TLS com registros de autenticação de canal:
   a) Evite que qualquer nome distinto (DN) abra um canal, usando um registro de autenticação de canal SSLPEERMAP para mapear todos os DNs para USERSRC(NOACCESS).

b) Permita que DNs específicos ou conjuntos de DNs abram um canal, usando um registro de autenticação de canal SSLPEERMAP para mapeá-los para USERSRC(CHANNEL).

2. Usando TLS com uma saída de segurança:
   a) Configure MCAUSER no canal de conexão do servidor para um identificador de usuários sem privilégios.
   b) Grave uma saída de segurança para designar um valor MCAUSER, dependendo do valor do DN TLS que ele recebe nos campos SSLPeerNamePtr e SSLPeerNameLength passados para a saída na estrutura MQCD.
3. Usando TLS com valores de definição de canal fixos:
   a) Configure SSLPEER no canal de conexão do servidor para um valor específico ou limite o intervalo de valores.
   b) Configure MCAUSER no canal de conexão do servidor para o ID do usuário com o qual o canal deve ser executado.
4. Usando registros de autenticação de canal em canais que não usam TLS:
   a) Evite que qualquer endereço IP abra canais, usando um registro de autenticação de canal de mapeamento de endereço com ADDRESS(*) e USERSRC(NOACCESS).
   b) Permita que endereços IP específicos abram canais, usando registros de autenticação de canal de mapeamento de endereço para esses endereços com USERSRC(CHANNEL).
5. Usando uma saída de segurança:
   a) Grave uma saída de segurança para autorizar conexões com base em qualquer propriedade escolhida, por exemplo, o endereço IP de origem.
6. Também é possível usar registros de autenticação de canal com uma saída de segurança ou usar todos os três métodos, se suas circunstâncias específicas exigirem.

Bloqueando Endereços IP Específicos
É possível evitar que um canal específico aceite uma conexão de entrada de um endereço IP, ou evitar que o gerenciador de filas inteiro permita o acesso a partir de um endereço IP, usando um registro de autenticação de canal.

Antes de começar

Ative registros de autenticação de canal executando o comando a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Sobre esta tarefa

Para desaprovar canais específicos para aceitação de uma conexão de entrada e assegurar que conexões sejam aceitas apenas ao usar o nome de canal correto, um tipo de regra pode ser usado para bloquear endereços IP. Para desaprovar um endereço IP a acessar o gerenciador de fias inteiro, você normalmente usaria um firewall para bloqueá-lo permanentemente. Entretanto, um outro tipo de regra pode ser usado para permitir que você bloqueie alguns endereços temporariamente, por exemplo, enquanto você estiver aguardando pela atualização do firewall.

Procedimento

- Para bloquear endereços IP do uso de um canal específico, configure um registro de autenticação de canal usando o comando MQSC **SET CHLAUTH**, ou o comando PCF **Set Channel Authentication Record**.

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

  Existem três partes para o comando:

SET CHLAUTH (*generic-channel-name*)
Use esta parte do comando para controlar se você deseja bloquear uma conexão para o gerenciador de filas inteiro, canal único ou intervalo de canais. O que você colocou aqui determina quais áreas estão cobertas.

Por exemplo:

- SET CHLAUTH('*') - bloqueia todos os canais em um gerenciador de filas, ou seja, todo o gerenciador de filas
- SET CHLAUTH('SYSTEM.*') - bloqueia todos os canais iniciados com SYSTEM.
- SET CHLAUTH('SYSTEM.DEF.SVRCONN') - bloqueia o canal SYSTEM.DEF.SVRCONN

Tipo de regra CHLAUTH
Use esta parte do comando para especificar o tipo de comando e determinar se você deseja fornecer um único endereço ou uma lista de endereços.

Por exemplo:

- TYPE(ADDRESSMAP) - use ADDRESSMAP se você deseja fornecer um único endereço ou um endereço curinga. Por exemplo, ADDRESS('192.168.*') bloqueia quaisquer conexões provenientes de um endereço IP com início no 192.168.

  Para obter mais informações sobre como filtrar endereços IP com padrões, consulte Endereços IP genéricos.
- TYPE(BLOCKADDR) - Use BLOCKADDR se você deseja fornecer uma lista de endereços para o bloqueio.

Parâmetros Adicionais
Esses parâmetros são dependentes do tipo de regra usada na segunda parte do comando:

- Para TYPE(ADDRESSMAP), use o ADDRESS
- Para TYPE(BLOCKADDR), use o ADDRLIST

Referências relacionadas
SET CHLAUTH

Bloqueando temporariamente endereços IP específicos se o gerenciador de filas não estiver em execução
Você pode desejar bloquear determinados endereços IP ou intervalos de endereços, quando o gerenciador de filas não estiver em execução e não é possível, portanto, emitir comandos MQSC. É possível bloquear temporariamente os endereços IP em uma base excepcional modificando o arquivo `blockaddr.ini`.

Sobre esta tarefa

O arquivo `blockaddr.ini` contém uma cópia das definições de BLOCKADDR que são usadas pelo Gerenciador de Filas. Esse arquivo é lido pelo listener se o listener é iniciado antes do gerenciador de filas. Nessas circunstâncias, o listener utiliza quaisquer valores que você tenha incluído manualmente no arquivo `blockaddr.ini`

No entanto, esteja ciente de que quando o Gerenciador de Filas é iniciado, ele grava o conjunto de definições de BLOCKADDR no arquivo `blockaddr.ini`, substituindo qualquer edição manual que você possa ter feito. Da mesma forma, sempre que você incluir ou excluir uma definição de BLOCKADDR usando o comando **SET CHLAUTH**, o arquivo `blockaddr.ini` é atualizado. É possível, portanto, tornar as mudanças permanentes para as definições de BLOCKADDR somente usando o comando **SET CHLAUTH** quando o gerenciador de filas estiver em execução.

Procedimento

1. Abra o arquivo `blockaddr.ini` em um editor de texto.
   O arquivo está localizado no diretório de dados do gerenciador de filas.
2. Inclua endereços IP como pares de valor de palavra-chave simples, em que a palavra-chave é `Addr`.

Para obter informações sobre filtrar endereços IP com padrões, consulte Endereços IP genéricos.

Por exemplo:

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

Tarefas relacionadas

"Bloqueando Endereços IP Específicos" na página 390

É possível evitar que um canal específico aceite uma conexão de entrada de um endereço IP, ou evitar que o gerenciador de filas inteiro permita o acesso a partir de um endereço IP, usando um registro de autenticação de canal.

Referências relacionadas

SET CHLAUTH

Bloqueando IDs de Usuários Específicos

É possível evitar que usuários específicos usem um canal especificando IDs de usuários que, se declarados, fazem com que o canal termine. Faça isso configurando um registro de autenticação de canal de canal.

Antes de começar

Certifique-se de que os registros de autenticação de canal são ativados como a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procedimento

Configure um registro de autenticação de canal usando o comando de MQSC **SET CHLAUTH**, ou o comando de PCF **Set Channel Authentication Record**. Por exemplo, é possível emitir o comando de MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.

A lista de usuários fornecida em um TYPE(BLOCKUSER) se aplica somente a canais SVRCONN e não a canais de gerenciador de filas para gerenciador de filas.

userID1 e *userID2* são cada um o ID de um usuário que deve ser evitado de usar o canal. Também é possível especificar o valor especial *MQADMIN para fazer referência a usuários administrativos privilegiados. Para obter informações adicionais sobre usuários privilegiados, consulte "Usuários Privilegiados" na página 325. Para obter informações adicionais sobre *MQADMIN, consulte SET CHLAUTH.

Referências relacionadas

SET CHLAUTH

Mapeando um Gerenciador de Filas Remoto para um ID do Usuário MCAUSER

É possível usar um registro de autenticação de canal para configurar o atributo MCAUSER de um canal de acordo com o gerenciador de filas a partir do qual o canal está conectando.

Antes de começar

Certifique-se de que os registros de autenticação de canal são ativados como a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Sobre esta tarefa

Como opção, é possível restringir os endereços IP aos quais a regra se aplica.

Observe que essa técnica não se aplica a canais de conexão do servidor. Se você especificar o nome de um canal de conexão do servidor nos comandos a seguir, isso não tem efeito.

Procedimento

- Configure um registro de autenticação de canal usando o comando de MQSC **SET CHLAUTH**, ou o comando de PCF **Set Channel Authentication Record**. Por exemplo, é possível emitir o comando de MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user)
```

  generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.
  generic-partner-qmgr-name é o nome do gerenciador de filas ou um padrão que inclui o símbolo de asterisco (*) como um curinga que corresponde ao nome do gerenciador de filas.
  user é o ID do usuário a ser usado para todas as conexões do gerenciador de filas especificado.

- Para restringir esse comando a determinados endereços IP, inclua o parâmetro **ADDRESS** da seguinte forma:

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user) ADDRESS(
generic-ip-address)
```

  generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.
  generic-ip-address é um endereço único ou um padrão que inclui o símbolo de asterisco (*) como um curinga ou o hífen (-) para indicar um intervalo, que corresponda ao endereço. Para obter informações adicionais sobre endereços IP genéricos, consulte Endereços IP genéricos.

Referências relacionadas
SET CHLAUTH

Mapeando um ID de usuário cliente para um ID do usuário MCAUSER
É possível usar um registro de autenticação de canal para mudar o atributo MCAUSER de um canal de conexão do servidor, de acordo com o ID do usuário recebido de um cliente.

Antes de começar

Certifique-se de que os registros de autenticação de canal são ativados como a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Sobre esta tarefa

Observe que essa técnica se aplica somente a canais de conexão do servidor. Não tem nenhum efeito em outros tipos de canais.

Procedimento

Configure um registro de autenticação de canal usando o comando MQSC **SET CHLAUTH**ou o comando PCF **Set Channel Authentication Record** . Por exemplo, é possível emitir o comando de MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
```

```
MCAUSER(
user)
```

generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.
client-user-name é o ID do usuário associado com a conexão de clientes, o valor poderia ser declarado pelo aplicativo cliente, alterado pela autenticação de conexão usando adoção antecipada ou configurado por meio de uma saída do canal.
user é o ID do usuário a ser usado em vez de o nome de usuário do cliente.

Referências relacionadas
SET CHLAUTH
Atributos da sub-rotina de canais (ChlauthEarlyAdopt)

Mapeando um Nome Distinto de SSL ou TLS para um ID do Usuário MCAUSER
É possível usar um registro de autenticação de canal para configurar o atributo MCAUSER de um canal de acordo com o Nome Distinto (DN) recebido.

Antes de começar
Certifique-se de que os registros de autenticação de canal são ativados como a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procedimento
Configure um registro de autenticação de canal usando o comando de MQSC **SET CHLAUTH**, ou o comando de PCF **Set Channel Authentication Record**. Por exemplo, é possível emitir o comando de MQSC:

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP)
SSLPEER(generic-ssl-peer-name) SSLCERTI(generic-issuer-name)
USERSRC(MAP) MCAUSER(user)
```

generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.
generic-ssl-peer-name é uma d que segue as regras de padrão do IBM MQ para valores de SSLPEER. Consulte Regras do IBM MQ para valores SSLPEER.
user é o ID do usuário a ser usado para todas as conexões usando o DN especificado.
generic-issuer-name refere-se ao DN do emissor do certificado para corresponder. Esse parâmetro é opcional mas é necessário usá-lo para evitar de corresponder de maneira falsa ao certificado errado, se várias autoridades de certificação estiverem em uso.

Referências relacionadas
SET CHLAUTH

Bloqueando Acesso de um Gerenciador de Filas Remotas
É possível usar um registro de autenticação de canal para evitar que um gerenciador de filas remotas inicie canais.

Antes de começar
Certifique-se de que os registros de autenticação de canal são ativados como a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Sobre esta tarefa

Observe que essa técnica não se aplica a canais de conexão do servidor. Se você especificar o nome de um canal de conexão do servidor no comando a seguir, ele não tem efeito.

Procedimento

Configure um registro de autenticação de canal usando o comando de MQSC **SET CHLAUTH**, ou o comando de PCF **Set Channel Authentication Record**. Por exemplo, é possível emitir o comando de MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(QMGRMAP) QMNAME(' generic-partner-qmgr-name ')
USERSRC(NOACCESS)
```

generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.
generic-partner-qmgr-name é o nome do gerenciador de filas ou um padrão que inclui o símbolo de asterisco (*) como um curinga que corresponde ao nome do gerenciador de filas.

Referências relacionadas

SET CHLAUTH

Bloqueando o acesso para um ID de usuário cliente

É possível usar um registro de autenticação de canal para evitar que um ID de usuário cliente estabeleça uma conexão de canal.

Antes de começar

Certifique-se de que os registros de autenticação de canal são ativados como a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Sobre esta tarefa

Observe que essa técnica se aplica somente a canais de conexão do servidor. Não tem nenhum efeito em outros tipos de canais.

Procedimento

Configure um registro de autenticação de canal usando o comando de MQSC **SET CHLAUTH**, ou o comando de PCF **Set Channel Authentication Record**. Por exemplo, é possível emitir o comando de MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(USERMAP) CLNTUSER(' client-user-name ')
USERSRC(NOACCESS)
```

generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.
client-user-name é o ID do usuário associado com a conexão de clientes, o valor poderia ser declarado pelo aplicativo cliente, alterado pela autenticação de conexão usando adoção antecipada ou configurado por meio de uma saída do canal.

Referências relacionadas

SET CHLAUTH

Bloqueando o acesso para um Nome Distinto SSL ou TLS

É possível usar um registro de autenticação de canal para evitar que um Nome Distinto (DN) TLS inicie canais.

Antes de começar

Certifique-se de que os registros de autenticação de canal são ativados como a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procedimento

Configure um registro de autenticação de canal usando o comando de MQSC **SET CHLAUTH**, ou o comando de PCF **Set Channel Authentication Record**. Por exemplo, é possível emitir o comando de MQSC:

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP)
SSLPEER('generic-ssl-peer-name') SSLCERTI(generic-issuer-name)
USERSRC(NOACCESS)
```

generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.

generic-ssl-peer-name é uma d que segue as regras de padrão do IBM MQ para valores de SSLPEER. Consulte Regras do IBM MQ para valores SSLPEER.

generic-issuer-name refere-se ao DN do emissor do certificado para corresponder. Esse parâmetro é opcional mas é necessário usá-lo para evitar de corresponder de maneira falsa ao certificado errado, se várias autoridades de certificação estiverem em uso.

Referências relacionadas

SET CHLAUTH

Mapeando um Endereço IP para um ID do Usuário MCAUSER

É possível usar um registro de autenticação de canal para configurar o atributo MCAUSER de um canal de acordo com o endereço IP a partir do qual a conexão é recebida.

Antes de começar

Certifique-se de que os registros de autenticação de canal são ativados como a seguir:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procedimento

Configure um registro de autenticação de canal usando o comando de MQSC **SET CHLAUTH**, ou o comando de PCF **Set Channel Authentication Record**. Por exemplo, é possível emitir o comando de MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(ADDRESSMAP) ADDRESS(' generic-ip-address ')
USERSRC(MAP) MCAUSER(user)
```

generic-channel-name é o nome de um canal para o qual você deseja acesso ao controle, ou um padrão incluindo o símbolo asterisco (*) como um curinga que corresponde ao nome de canal.

user é o ID do usuário a ser usado para todas as conexões usando o DN especificado.

generic-ip-address é o endereço a partir do qual a conexão está sendo feita ou um padrão que inclui o asterisco (*) como um curinga ou o hífen (-) para indicar um intervalo, que corresponda ao endereço.

Referências relacionadas

SET CHLAUTH

Desativando o Acesso Remoto ao Gerenciador de Filas

Para que os aplicativos clientes não sejam conectados ao gerenciador de filas, desative o acesso remoto a eles.

Sobre esta tarefa

Evite que aplicativos clientes sejam conectados ao gerenciador de filas em uma das seguintes maneiras:

Procedimento

- Exclua todos os canais de conexão do servidor usando o comando **DELETE CHANNEL**do MQSC
- Configure o identificador de usuários do agente do canal de mensagens (MCAUSER) do canal com um ID do usuário sem direitos de acesso, usando o comando MQSC **ALTER CHANNEL**.

Configurando a Segurança de Conexão

Conceda a autoridade de conexão com o gerenciador de filas a cada usuário ou grupo de usuários que tiver uma necessidade de negócios.

Sobre esta tarefa

Para configurar a segurança de conexão, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW

  No AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  ```

- IBM i

  No IBM i:

  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
  ```

- z/OS

  No z/OS:

  ```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
  PERMIT QMgrName.BATCH CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.CICS UACC(NONE)
  PERMIT QMgrName.CICS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.IMS UACC(NONE)
  PERMIT QMgrName.IMS CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
  PERMIT QMgrName.CHIN CLASS(MQCONN) ID(GroupName) ACCESS(READ)
  ```

  Estes comandos concedem autoridade para conectar-se para lote, CICS, IMS e o inicializador de canais (CHIN). Se você não usar um tipo particular de conexão, omita os comandos relevantes.

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Conceitos relacionados

“Connection security profiles for the channel initiator” na página 206

Profiles for checking connections from the channel initiator are composed of the queue manager or queue sharing group name followed by the word *CHIN*. Give the user ID used by the channel initiator started task address space READ access to the connection profile.

Controlando o Acesso de Usuário a Filas

Você deseja controlar o acesso ao aplicativo a filas. Use este tópico para determinar quais ações executar.

Para cada instrução verdadeira na primeira coluna, execute a ação indicada na segunda coluna.

Declaração	**Ação**
O aplicativo obtém mensagens de uma fila	Consulte "Concedendo autoridade para obter mensagens de filas" na página 398
O aplicativo configura contexto	Consulte "Concedendo autoridade para configurar o contexto" na página 399
O aplicativo passa contexto	Consulte "Concedendo autoridade para passar o contexto" na página 400
O aplicativo coloca mensagens em uma fila armazenada em cluster	Consulte "Autorizando a Colocação de Mensagens em Filas de Cluster Remotas" na página 486
O aplicativo coloca mensagens em uma fila local	Consulte "Concedendo autoridade para colocar mensagens em uma fila local" na página 401
O aplicativo coloca mensagens em uma fila modelo	Consulte "Concedendo autoridade para colocar mensagens em uma fila modelo" na página 401
O aplicativo coloca mensagens em uma fila remota	Consulte "Concedendo autoridade para colocar mensagens em uma fila do cluster remoto" na página 402

Concedendo autoridade para obter mensagens de filas

Conceda a autoridade para obter mensagens de uma fila ou um conjunto de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para obter mensagens de algumas filas, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- Windows
  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
  ```

- IBM i
  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME(' QMgrName ')
  ```

- z/OS
  Para z/OS, emita os seguintes comandos:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas. Em z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

Concedendo autoridade para configurar o contexto
Conceda a autoridade para configurar contexto em uma mensagem que está sendo colocada, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para configurar contexto em algumas filas, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  Para sistemas AIX, Linux, and Windows, emita um dos comandos a seguir:
  - Para configurar apenas contexto de identidade:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
    ```
  - Para configurar todo o contexto:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setall
    ```

  Nota: Para usar a autoridade `setid` ou `setall`, as autorizações devem ser concedidas tanto no objeto de filas apropriado como também no objeto de gerenciador de filas.
- IBM i
  Para o IBM i, emita um dos comandos a seguir:
  - Para configurar apenas contexto de identidade:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME('
    QMgrName ')
    ```
  - Para configurar todo o contexto:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('
    QMgrName ')
    ```
- z/OS
  Para o z/OS, emita um dos conjuntos de comandos a seguir:
  - Para configurar apenas contexto de identidade:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

- Para configurar todo o contexto:

```
RDEFINE MQQUEUE QMgrName. ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

Concedendo autoridade para passar o contexto
Conceda a autoridade para passar contexto de uma mensagem recuperada para uma que está sendo colocada, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para passar contexto em algumas filas, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  Para sistemas AIX, Linux, and Windows, emita um dos comandos a seguir:
  - Para passar apenas contexto de identidade:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
```

  - Para passar todo o contexto:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
```

- IBM i
  Para o IBM i, emita um dos comandos a seguir:
  - Para passar apenas contexto de identidade:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('
QMgrName ')
```

  - Para passar todo o contexto:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('
QMgrName ')
```

- z/OS
  Para o z/OS, emita os comandos a seguir para transmitir o contexto de identidade ou todo o contexto:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

Concedendo autoridade para colocar mensagens em uma fila local
Conceda a autoridade para colocar mensagens em uma fila local ou um conjunto de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para colocar mensagens em algumas filas locais, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
  ```

- IBM i
  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
  ```

- z/OS
  Para z/OS, emita os seguintes comandos:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo autoridade para colocar mensagens em uma fila modelo
Conceda a autoridade para colocar mensagens em uma fila modelo ou um conjunto de filas modelo, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Filas modelo são usadas para criar filas dinâmicas. Você deve, portanto, conceder autoridade para ambas as filas, modelo e dinâmica. Para conceder essas autoridades, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW

  Para sistemas AIX, Linux, and Windows, emita os comandos a seguir:

  ```
  setmqaut -m QMgrName -n ModelQueueName -t queue -g GroupName +put
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
  ```

- IBM i

  Para IBM i, emita os seguintes comandos:

  ```
  GRTMQMAUT OBJ(' ModelQueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
  ```

- z/OS

  Para z/OS, emita os seguintes comandos:

  ```
  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)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ModelQueueName
  O nome da fila modelo em que as filas dinâmicas se baseiam.

  ObjectProfile
  O nome do perfil da fila dinâmica ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo autoridade para colocar mensagens em uma fila do cluster remoto
Conceda a autoridade para colocar mensagens em uma fila cluster remoto ou um conjunto de filas a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para colocar uma mensagem em uma fila do cluster remoto, é possível colocá-la em uma definição local de uma fila remota ou uma fila remota completa. Se você estiver usando uma definição local de uma fila remota, você precisa de autoridade para colocar o objeto local: consulte “Concedendo autoridade para colocar mensagens em uma fila local” na página 401. Se você estiver usando uma fila remota completa, você precisa de autoridade para colocar a fila remota. Conceda esta autoridade usando os comandos apropriados para seu sistema operacional.

O comportamento padrão é realizar o controle de acesso no SYSTEM.CLUSTER.TRANSMIT.QUEUE. Observe que esse comportamento se aplica mesmo se você estiver usando diversas filas de transmissão.

O comportamento específico descrito neste tópico se aplica apenas quando você tiver configurado o atributo **ClusterQueueAccessControl** no arquivo qm.ini para ser *RQMNome*, conforme descrito no tópico Sub-rotina de segurança, e tiver reiniciado o gerenciador de filas.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW

  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -t rqmname -n
  ObjectProfile -g GroupName +put
  ```

  Observe que é possível usar o objeto *rqmname* somente para filas de cluster remoto.

- IBM i

  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJTYPE(*RMTMQMNAME) OBJ('
  ObjectProfile') USER(GroupName) AUT(*PUT) MQMNAME('
  QMgrName')
  ```

  Observe que é possível usar o objeto RMTMQMNAME somente para filas de cluster remoto.

- z/OS

  Para z/OS, emita os seguintes comandos:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  Observe que é possível usar o nome do gerenciador de filas remotas (ou grupo de filas compartilhadas) somente para filas de clusters remotos.

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do gerenciador de filas remotas ou do perfil genérico para o qual mudar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Controlando o Acesso de Usuário aos Tópicos

É necessário controlar o acesso de aplicativos aos tópicos. Use este tópico para determinar quais ações executar.

Para cada instrução verdadeira na primeira coluna, execute a ação indicada na segunda coluna.

Tabela 74. Controlando o Acesso de Usuário aos Tópicos

Declaração	**Ação**
O aplicativo publica mensagens em um tópico	Consulte “Concedendo autoridade para publicar mensagens em um tópico” na página 404
O aplicativo assina um tópico	Consulte “Concedendo autoridade para assinar tópicos” na página 404

Concedendo autoridade para publicar mensagens em um tópico
Conceda a autoridade para publicar mensagens em um tópico ou um conjunto de tópicos, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para publicar mensagens em alguns tópicos, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
  ```

- IBM i
  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('
  QMgrName ')
  ```

- z/OS
  Para z/OS, emita os seguintes comandos:

  ```
  RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo autoridade para assinar tópicos
Conceda a autoridade para assinar um tópico ou um conjunto de tópicos, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para assinar alguns tópicos, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- IBM i

  Para o IBM i, emita o comando a seguir:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('
QMgrName ')
```

- z/OS

  Para z/OS, emita os seguintes comandos:

```
RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo autoridade para consultar em um gerenciador de filas

Conceda a autoridade para consultar em um gerenciador de filas, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para consultar em um gerenciador de filas, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW

  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- IBM i

  Para o IBM i, emita o comando a seguir:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME(' QMgrName
')
```

- z/OS

  Para z/OS, emita os seguintes comandos:

```
RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Esses comandos concedem acesso ao gerenciador de filas especificado. Para permitir que o usuário use o comando MQINQ, emita os seguintes comandos:

```
RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Os nomes das variáveis possuem os seguintes significados:

QMgrName
O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

ObjectProfile
O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

GroupName
O nome do grupo que receberá acesso.

Concedendo autoridade para acessar processos

Conceda a autoridade para acessar um processo ou um conjunto de processos, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para acessar alguns processos, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW
  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
  ```

- IBM i
  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME(' QMgrName
  ')
  ```

- z/OS
  Para z/OS, emita os seguintes comandos:

  ```
  RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

Concedendo autoridade para acessar listas de nomes

Conceda a autoridade para acessar uma lista de nomes ou um conjunto de listas de nomes, a cada grupo de usuários com uma necessidade de negócios para isso.

Sobre esta tarefa

Para conceder a autoridade para acessar algumas listas de nomes, use os comandos apropriados de seu sistema operacional.

Multi Em Multiplataformas, também é possível utilizar o comando SET AUTHREC

Procedimento

- ALW

  Para sistemas AIX, Linux, and Windows, emita o comando a seguir:

  ```
  setmqaut -m QMgrName -n
  ObjectProfile -t namelist -g GroupName
  +all
  ```

- IBM i

  Para o IBM i, emita o comando a seguir:

  ```
  GRTMQMAUT OBJ('ObjectProfile
  ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
  QMgrName')
  ```

- z/OS

  Para z/OS, emita os seguintes comandos:

  ```
  RDEFINE MQNLIST
  QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile
  CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
  ```

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  ObjectProfile
  O nome do perfil de objeto ou do perfil genérico do qual alterar as autorizações.

  GroupName
  O nome do grupo que receberá acesso.

ALW Autoridade para administrar o IBM MQ no AIX, Linux, and Windows

Os administradores do IBM MQ podem usar todos os comandos do IBM MQ e conceder autoridades para outros usuários. Quando administradores emitem comandos para gerenciadores de filas remotas, ele devem ter a autoridade necessária no gerenciador de filas remotas. Considerações adicionais se aplicam aos sistemas Windows.

Os administradores do IBM MQ têm autoridade para usar todos os comandos do IBM MQ (incluindo os comandos para conceder autoridades do IBM MQ para outros usuários).

Para ser um administrador do IBM MQ, deve-se ser membro de um grupo especial denominado grupo **mqm**.

Windows Como alternativa, apenas no Windows , as contas locais podem administrar o IBM MQ se forem membros do grupo de Administradores em sistemas Windows .

Atenção: É possível incluir seu usuário do Azure AD no grupo mqm usando um comando de administrador. Por exemplo, use o comando `net localgroup mqm AzureAD\<your userID> /add`. Em seguida, execute comandos de administração do IBM MQ ou use o IBM MQ Explorer.

O grupo **mqm** é criado automaticamente quando o IBM MQ está instalado. É possível incluir usuários adicionais no grupo para permitir que eles executem a administração. Todos os membros desse grupo possuem acesso a todos os recursos. Esse acesso pode ser revogado somente removendo um usuário do grupo **mqm** e emitindo o comando **REFRESH SECURITY**.

Os administradores podem usar comandos de controle para administrar o IBM MQ. Um desses comandos de controle é **setmqaut**, que é usado para conceder autoridades a outros usuários para permitir que eles acessem ou controlem os recursos do IBM MQ. Os comandos PCF para gerenciar os registros de autoridade estão disponíveis a não administradores aos quais tenham sido concedidas autoridades dsp e chg no gerenciador de filas. Para obter mais informações sobre o gerenciamento de autoridades usando comandos PCF, veja Formatos de comando programável.

Os administradores devem ter as autoridades requeridas para os comandos MQSC serem processados pelo gerenciador de fila remoto. O IBM MQ Explorer emite comandos PCF (formato de comando programável) para executar tarefas de administração. Os administradores não requerem autoridades adicionais para usar o IBM MQ Explorer para administrar um gerenciador de filas no sistema local. Quando o IBM MQ Explorer é usado para administrar um gerenciador de filas em outro sistema, os administradores devem ter as autoridades necessárias para que os comandos PCF sejam processados pelo gerenciador de filas remoto.

Atenção: Não é necessário ser um administrador para usar o comando control **runmqsc**, que emite comandos IBM MQ Script (MQSC).

Quando **runmqsc** é utilizado no modo indireto para enviar comandos MQSC para um gerenciador de fila remoto, cada comando será encapsulado dentro de um comando PCF Escape.

Para obter mais informações sobre verificações de autoridade quando os comandos PCF e MQSC são processados, consulte os seguintes tópicos:

- Para comandos PCF que operam em gerenciadores de filas, filas, processos, listas de nomes e objetos de informações sobre autenticação, consulte Autoridade para trabalhar com objetos do IBM MQ. Consulte esta seção para obter os comandos MQSC equivalentes encapsulados nos comandos PCF Escape.
- Para comandos PCF que operam em canais, iniciadores de canais, listeners e clusters, consulte Segurança de Canal.
- Para comandos PCF que operam em registros de autoridade, consulte verificação de autoridade para comandos PCF
- z/OS Para comandos MQSC que são processados pelo servidor de comando no IBM MQ for z/OS, consulte Segurança do comando e segurança de recurso do comando no z/OS .

Além disso, em sistemas Windows , a conta SYSTEM tem acesso total aos recursos do IBM MQ

Em plataformas AIX and Linux, um ID do usuário especial de **mqm** também é criado, para uso somente pelo produto. Ele nunca deverá estar disponível para usuários não privilegiados. Todos os objetos do IBM MQ são de propriedade do ID do usuário **mqm**.

Em sistemas Windows , os membros do grupo de Administradores também podem administrar qualquer gerenciador de filas, assim como a conta SYSTEM Também é possível criar um grupo **mqm** de domínio no controlador de domínio que contenha todos os IDs de usuários privilegiados ativos dentro do domínio e incluí-lo no grupo **mqm** local. Alguns comandos, por exemplo **crtmqm**, manipulam as autoridades nos objetos IBM MQ e, portanto, precisam de autoridade para trabalhar com esses objetos (conforme descrito nas seções a seguir). Os membros do grupo **mqm** têm autoridade para trabalhar com todos os objetos, mas pode haver circunstâncias nos sistemas Windows em que a autoridade será negada se você tiver um

usuário local e um usuário autenticado pelo domínio com o mesmo nome. Isso é descrito no “Principais e grupos no AIX, Linux, and Windows” na página 412.

Versões do Windows com um recurso Controle de Conta do Usuário (UAC) restringem as ações que os usuários podem executar em certos recursos do sistema operacional, mesmo que sejam membros do grupo Administradores. Se o seu ID do usuário está no grupo Administradores, mas não no grupo **mqm**, deve-se usar um prompt de comandos elevado para emitir comandos administrativos do IBM MQ, como **crtmqm**, caso contrário, o erro `AMQ7077: você não está autorizado a executar a operação solicitada` é gerado. Para abrir um prompt de comandos elevado, clique com o botão direito no item de menu iniciar ou ícone para o prompt de comandos e selecione **Executar como administrador**.

Você não precisa ser um membro do grupo **mqm** para executar as ações a seguir:

- Emita os comandos a partir de um programa de aplicativo que emite os comandos PCF ou comandos MQSC em um comando Escape PCF, a menos que os comandos manipulem os iniciadores de canal. (Esses comandos são descritos em “Protegendo Definições do Inicializador de Canais” na página 121).
- Emita as chamadas MQI a partir de um programa de aplicativo (a menos que você queira usar as ligações de caminho rápido na chamada `MQCONNX`).
- Use o comando `crtmqcvx` para criar um fragmento de código que executa conversão de dados em estruturas de tipo de dados.
- Usar o comando `dspmq` para exibir gerenciadores de filas.
- Use o comando `dspmqtrc` para exibir a saída de rastreio formatado do IBM MQ.

Uma limitação de 12 caracteres aplica-se ao grupo e aos IDs do usuário.

As plataformas UNIX and Linux geralmente restringem o comprimento de um ID do usuário a 12 caracteres. O AIX 5.3 aumentou esse limite, mas o IBM MQ ainda observa uma restrição de 12 caracteres em todas as plataformas UNIX and Linux. Se você usar um ID do usuário com mais de 12 caracteres, o IBM MQ substitui-o com o valor de `UNKNOWN`. Não defina um ID do usuário com um valor de `UNKNOWN`.

ALW Gerenciando o grupo mqm no AIX, Linux, and Windows

Usuários no grupo mqm são concedidos privilégios administrativos completos sobre o IBM MQ. Por esta razão, não é necessário inscrever usuários comuns e de aplicativos no grupo mqm. O grupo mqm deve conter contas somente dos administradores do IBM MQ.

Estas tarefas estão descritas em:

- Windows Criando e gerenciando grupos no Windows
- AIX Criando e gerenciando grupos no AIX
- Linux Criando e gerenciando grupos no Linux

Windows Se o seu controlador de domínio é executado no Windows 2000 ou Windows 2003 ou mais recente, seu administrador de domínio pode ter que configurar uma conta especial para o IBM MQ usar. Para obter mais informações, consulte Configurando o IBM MQ com o Prepare IBM MQ Wizard e Criando e configurando contas de domínio do Windows para o IBM MQ.

ALW Autoridade para trabalhar com objetos do IBM MQ no AIX, Linux, and Windows

Todos os objetos são protegidos pelo IBM MQ e deve ser concedida autoridade apropriada aos diretores para acessá-los. Diferentes principais precisam de diferentes direitos de acesso a diferentes objetos.

Gerenciadores de filas, filas, definições de processos, listas de nomes, canais, canais de conexão do cliente, listeners, serviços e objetos de informações sobre autenticação são todos acessados a partir de aplicativos que usam chamadas MQI ou comandos PCF. Esses são todos os recursos protegidos pelo IBM MQ e os aplicativos precisam ser concedida permissão para acessá-los. A entidade que faz a solicitação

pode ser um usuário, um programa de aplicativo que emite uma chamada MQI ou um programa de administração que emite um comando PCF. O identificador do solicitante é referido como o *principal*.

Diferentes grupos de principais podem receber diferentes tipos de autoridade de acesso ao mesmo objeto. Por exemplo, para uma fila específica, um grupo pode ter permissão para executar ambas as operações put e get; outro grupo pode ter permissão somente para navegar pela fila (MQGET com a opção de navegação). De forma semelhante, alguns grupos podem ter as autoridades put e get para uma fila, mas podem não ter permissão para alterar atributos da fila ou para excluí-la.

Algumas operações são especialmente sigilosas e devem ser limitadas a usuários privilegiados. Por exemplo:

- Acesso a algumas filas especiais, como filas de transmissão ou a fila de comandos SYSTEM.ADMIN.COMMAND.QUEUE
- Execução de programas que usam opções de contexto completas de MQI
- Criação e exclusão de filas de aplicativos

Permissão de acesso completo a um objeto é determinada automaticamente para o ID do usuário que criou o objeto e para todos os membros do grupo mqm (e aos membros do grupo Administradores local em sistemas Windows).

Conceitos relacionados

"Autoridade para administrar o IBM MQ no AIX, Linux, and Windows" na página 407
Os administradores do IBM MQ podem usar todos os comandos do IBM MQ e conceder autoridades para outros usuários. Quando administradores emitem comandos para gerenciadores de filas remotas, ele devem ter a autoridade necessária no gerenciador de filas remotas. Considerações adicionais se aplicam aos sistemas Windows.

ALW Quando verificações de segurança são feitas no AIX, Linux, and Windows

As verificações de segurança são feitas geralmente ao conectar-se a um gerenciador de filas, abrir ou fechar objetos e colocar ou obter mensagens.

As verificações de segurança feitas para um aplicativo típico são as seguintes:

Conectando-se ao gerenciador de filas (chamadas MQCONN ou MQCONNX)

Eta é a primeira vez que o aplicativo é associado a um determinado gerenciador de filas. O gerenciador de filas interroga o ambiente operacional para descobrir o ID do usuário associado ao aplicativo. O IBM MQ, em seguida, verifica se o ID do usuário está autorizado a se conectar ao gerenciador de filas e retém o ID do usuário para verificações futuras.

Os usuários não têm de efetuar sign on para o IBM MQ; o IBM MQ assume que os usuários efetuaram sign on ao sistema operacional subjacente e foram autenticados por ele.

Abrindo o objeto (chamadas MQOPEN ou MQPUT1)

Os objetos do IBM MQ são acessados abrindo o objeto e emitindo comandos com relação a ele. Todas as verificações de recursos são executadas quando o objeto é aberto, em vez de quando ele é realmente acessado. Isso significa que a solicitação de **MQOPEN** deve especificar o tipo de acesso necessário (por exemplo, se o usuário deseja apenas pesquisar o objeto ou executar uma atualização, como colocar mensagens em uma fila).

O IBM MQ verifica o recurso que é nomeado na solicitação **MQOPEN**. Para um objeto de fila de alias ou fila remota, a autorização usada é aquela do próprio objeto, não da fila à qual a fila do alias ou a fila remota é resolvida. Isso significa que o usuário não precisa de permissão para acessá-la. Limite a autoridade para criar filas a usuários privilegiados. Se você não fizer isto, os usuários podem efetuar bypass do controle de acesso normal simplesmente criando um alias. Se uma fila remota for referida explicitamente com ambos os nomes, da fila e do gerenciador de filas, a fila de transmissão associada ao gerenciador de filas remotas será verificada.

A autoridade para uma fila dinâmica baseia-se naquela da fila modelo da qual se deriva, mas não é necessariamente a mesma. Isso é descrito na Nota "1" na página 140.

O ID do usuário usado pelo gerenciador de filas para verificações de acesso é o ID do usuário obtido do ambiente operacional do aplicativo conectado ao gerenciador de filas. Um aplicativo adequadamente autorizado pode emitir uma chamada **MQOPEN** especificando um ID do usuário alternativo; as verificações de controle de acesso são então feitas no ID do usuário alternativo. Isso não altera o ID do usuário associado ao aplicativo, apenas aquele usado para verificações de controle de acesso.

Colocando e obtendo mensagens (chamadas MQPUT ou MQGET)
Não são executadas verificações de controle de acesso.

Fechando o objeto (MQCLOSE)
Não são executadas verificações de controle de acesso, a menos que o **MQCLOSE** resulte na exclusão de uma fila dinâmica. Neste caso, há uma verificação para ver se o ID do usuário tem autorização para excluir a fila.

Inscrevendo-se em um tópico (MQSUB)
Quando um aplicativo é subscrito para um tópico, ele especifica o tipo de operação que precisa executar. Ele estará criando uma nova assinatura, alterando uma assinatura existente ou continuando uma assinatura existente sem alterá-la. Para cada tipo de operação, o gerenciador de filas verifica se o ID do usuário que está associado ao aplicativo possui a autoridade para executar a operação.

Quando um aplicativo é subscrito para um tópico, as verificações de autoridade são executadas com relação aos objetos de tópicos que estão localizados na árvore de tópicos ou acima do ponto na árvore de tópicos no qual a aplicativo foi subscrito. As verificações de autoridade podem envolver verificações em mais de um objeto de tópico.

O ID do usuário que o gerenciador de fila utiliza para as verificações de autoridade é o ID obtido do sistema operacional quando o aplicativo estabelece conexão com o gerenciador de fila.

O gerenciador de filas executa verificações de autoridade em filas de assinantes, mas não em filas gerenciadas.

ALW Como o controle de acesso é implementado pelo IBM MQ no AIX, Linux, and Windows

O IBM MQ usa os serviços de segurança fornecidos pelo sistema operacional subjacente, usando o gerenciador de autoridade de objeto. O IBM MQ fornece comandos para criar e manter listas de controle de acesso.

Uma interface de controle de acesso chamada Interface de serviço de autorização é parte do IBM MQ. O IBM MQ fornece uma implementação de um gerenciador de controle de acesso (em conformidade com a Interface de serviço de autorização) conhecido como o *gerenciador de autoridade de objeto (OAM)*. Isso é automaticamente instalado e ativado para cada gerenciador de filas que você criar, a menos que você especifique de outra maneira, (conforme descrito em "Evitando verificações de acesso de segurança nos sistemas AIX, Linux, and Windows" na página 369). O OAM pode ser substituído por qualquer usuário ou componente gravado do fornecedor que esteja em conformidade com a Interface de serviço de autorização.

O OAM explora os recursos de segurança do sistema operacional subjacente, usando IDs de usuário e de grupo do sistema operacional. Os usuários podem acessar objetos do IBM MQ somente se eles tiverem a autoridade correta. "Controlando o acesso a objetos usando o OAM no AIX, Linux, and Windows" na página 359 descreve como conceder e revogar essa autoridade.

O OAM mantém uma lista de controle de acesso (ACL) para cada recurso que controla. Os dados de autorização são armazenados em uma fila local chamada SYSTEM.AUTH.DATA.QUEUE. O acesso a essa fila é restrito a usuários no grupo mqm e, adicionalmente, no Windows, a usuários no grupo Administradores e usuários com login efetuado com o ID SYSTEM. O acesso de usuário à fila não pode ser alterado.

O IBM MQ fornece comandos para criar e manter listas de controle de acesso. Para obter informações adicionais sobre esses comandos, consulte "Controlando o acesso a objetos usando o OAM no AIX, Linux, and Windows" na página 359.

O IBM MQ transmite o OAM uma solicitação contendo um diretor, um nome de recurso e um tipo de acesso. O OAM concede ou rejeita o acesso com base na ACL que mantém. O IBM MQ segue a decisão do OAM; se o OAM não puder tomar uma decisão, o IBM MQ não permitirá o acesso.

ALW Identificando o ID do usuário no AIX, Linux, and Windows

O gerenciador de autoridade de objeto identifica o diretor que está solicitando acesso a um recurso. O ID do usuário usado como o principal varia de acordo com o contexto.

O gerenciador de autoridade de objeto (OAM) deve ser capaz de identificar quem está solicitando acesso a um recurso específico. O IBM MQ usa o termo *diretor* para fazer referência a este identificador. O principal é estabelecido quando o aplicativo se conecta ao gerenciador de filas pela primeira vez; ele é determinado pelo gerenciador de filas a partir do ID do usuário associado ao aplicativo de conexão. (Se o aplicativo emitir chamadas XA sem se conectar ao gerenciador de filas, o ID do usuário associado ao aplicativo que emite a chamada xa_open será usado para verificações de autoridade pelo gerenciador de filas.)

Em sistemas AIX and Linux, as rotinas de autorização verificam o ID do usuário real (efetuado log in) ou o ID do usuário efetivo associado ao aplicativo. O ID do usuário verificado pode depender do tipo de ligação. Para obter detalhes, consulte Serviços Instaláveis.

O IBM MQ propagará o ID do usuário recebido do sistema no cabeçalho da mensagem (estrutura do MQMD) de cada mensagem como a identificação do usuário. Esse identificador faz parte das informações de contexto da mensagem e é descrito em “Autoridade de contexto no AIX, Linux, and Windows” na página 415. Os aplicativos não podem alterar essas informações, a não ser que tenham sido autorizados a alterar as informações de contexto.

ALW *Principais e grupos no AIX, Linux, and Windows*

Principais podem pertencer a grupos. Concedendo acesso de recurso a grupos em vez de indivíduos, é possível reduzir a quantidade de administração requeridos. As Listas de controle de acesso (ACLs) baseiam-se em ambos os grupos e IDs de usuário.

Por exemplo, é possível definir um grupo consistindo em usuários que desejam executar um determinado aplicativo. Outros usuários podem receber acesso a todos os recursos que precisam, incluindo seus IDs de usuário no grupo apropriado.

Este processo de definição e gerenciamento de grupos está descrito para plataformas específicas:

- AIX Criando e gerenciando grupos no AIX
- Linux Criando e gerenciando grupos no Linux
- Windows Criando e gerenciando grupos no Windows

Um principal pode pertencer a mais de um grupo (seu conjunto de grupos). Ele tem a agregação de todas as autoridades concedidas a cada grupo em seu conjunto de grupos. Essas autoridades são armazenadas em cache, portanto, quaisquer mudanças feitas na associação ao grupo do principal não são reconhecidas até que o gerenciador de filas seja reiniciado, a menos que você emita o comando MQSC **REFRESH SECURITY** (ou seu equivalente PCF).

Linux AIX **SistemasAIX and Linux**

As listas de controle de acesso (ACLs) são baseadas em IDs de usuário e grupos e é possível usar para autorização configurando o atributo **SecurityPolicy** para o valor apropriado, conforme descrito na sub-rotina Serviço do arquivo qm.ini.

É possível usar o *modelo baseado em usuário* para autorização e isso permite que você use usuários e grupos. No entanto, ao especificar um usuário no comando setmqaut, as novas permissões se aplicam somente a esse usuário e não os grupos aos quais esse usuário pertence. Para obter informações adicionais, consulte “Permissões baseadas em usuário do OAM no AIX and Linux” na página 359.

Ao usar o *modelo baseado em grupo* para autorização, o grupo primário ao qual o ID do usuário pertence é incluído na ACL. O ID do usuário individual não é incluído e autoridade é concedida a

todos os membros desse grupo. Por causa disso, observe que é possível, acidentalmente, mudar a autoridade de um diretor mudando a autoridade de outro diretor no mesmo grupo.

Todos os usuários são nominalmente designados ao grupo de usuários padrão nobody e, por padrão, nenhuma autorização é fornecida a esse grupo. É possível mudar a autorização no grupo nobody para conceder acesso a recursos do IBM MQ para usuários sem as autorizações específicas.

DeIBM MQ 9.3.0 , você pode usar o`UserExternal` opção do**SecurityPolicy** atributo para criar um nome de usuário que não seja do sistema operacional. Se você criar um nome de usuário que não seja do sistema operacional, esse usuário não será considerado pertencente a nenhum grupo, exceto o grupo nobody. Para obter mais informações sobre essa opção, consulte crtmqm e Sub-rotina de serviço do arquivo qm.ini.

Não defina um ID do usuário com o valor UNKNOWN. O valor UNKNOWN é usado quando um ID do usuário é muito longo, portanto, os IDs de usuário arbitrários usariam as autoridades de acesso de UNKNOWN.

Consulte “Configurando autorizações” na página 421 para obter informações sobre o uso do LDAP.

Os IDs de usuário podem conter até 12 caracteres e nomes de grupos com até 12 caracteres.

Windows **SistemasWindows**

As ACLs baseiam-se em IDs de usuário e grupos. As verificações são as mesmas para o AIX and Linux. É possível ter usuários diferentes em domínios diferentes com o mesmo ID do usuário. O IBM MQ permite que os IDs de usuário sejam qualificados por um nome de domínio para que estes usuários possam receber diferentes níveis de acesso.

O nome do grupo pode, opcionalmente, incluir um nome de domínio, especificado nos formatos a seguir:

```
GroupName@domain domain_name\group_name
```

Grupos globais são marcadas pelo OAM somente em dois casos:

1. A sub-rotina de segurança do gerenciador de filas inclui a configuração: `GroupModel=GlobalGroups`. Consulte Segurança.
2. O gerenciador de filas está usando um grupo de acesso de segurança alternativo. Consulte **crtmqm** .

Os IDs de usuário podem conter até 20 caracteres, nomes de domínio até 15 caracteres e nomes de grupos até 64 caracteres.

O OAM verifica, primeiramente, o banco de dados de segurança local, em seguida, o banco de dados de domínio primário e, finalmente, o banco de dados de qualquer domínio confiável. O primeiro ID do usuário encontrado é usado pelo OAM para verificação. Cada um desses IDs de usuário pode ter associações diferentes ao grupo em um determinado computador.

Alguns comandos de controle (por exemplo, **crtmqm**) mudam autoridades em objetos do IBM MQ usando o gerenciador de autoridade de objeto (OAM). O OAM procura nos banco de dados de segurança na ordem fornecida no parágrafo precedente para determinar os direitos de autoridade de um determinado ID do usuário. Como resultado, a autoridade determinada pelo OAM pode substituir o fato de que um ID do usuário é um membro do grupo mqm local. Por exemplo, se você emitir o comando **crtmqm** a partir de um ID do usuário autenticado por um controlador de domínio que tenha associação do grupo mqm local por meio de um grupo global, o comando falhará se o sistema tiver um usuário local com o mesmo nome que não está no grupo mqm local.

Para obter mais informações sobre como configurar o atributo **SecurityPolicy** em Windows, consulte Sub-rotina de serviço do arquivo qm.ini

Windows *Identificadores de segurança (SIDs) do Windows*

O IBM MQ no Windows usa o SID no local onde ele está disponível. Se um SID do Windows não for fornecido com um pedido de autorização, o IBM MQ identifica o usuário com base no nome do usuário sozinho, mas isso pode resultar na autoridade errada sendo concedida.

Nos sistemas Windows, o identificador de segurança (SID) é usado para completar o ID do usuário. O SID contém informações que identificam os detalhes completos da conta do usuário no banco de dados do gerente de contas de segurança (SAM) do Windows no qual o usuário está definido. Quando uma mensagem é criada no IBM MQ for Windows, o IBM MQ armazena o SID no descritor de mensagens. Quando o IBM MQ no Windows executa verificações de autorização, ele usa o SID para consultar as informações completas do banco de dados do SAM. (O banco de dados do SAM em que o usuário está definido deve estar acessível para que essa consulta seja bem-sucedida.)

Por padrão, se um SID do Windows não for fornecido com um pedido de autorização, o IBM MQ identificará o usuário com base no nome do usuário sozinho. Ele faz isso procurando nos bancos de dados de segurança na seguinte ordem:

1. O banco de dados de segurança local
2. O banco de dados de segurança do domínio primário
3. O banco de dados de segurança de domínios confiáveis

Se o nome do usuário não for exclusivo, a autoridade incorreta do IBM MQ pode ser concedida. Para evitar esse problema, inclua um SID em cada pedido de autorização; o SID é usado pelo IBM MQ para estabelecer credenciais do usuário.

Para especificar que todas as solicitações de autorização devem incluir um SID, use **regedit**. Configure SecurityPolicy como NTSIDsRequired.

ALW Autoridade de usuário alternativo no AIX, Linux, and Windows

É possível especificar que um ID do usuário possa usar a autoridade de outro usuário quando acessar um objeto do IBM MQ. Isso é denominado *autoridade de usuário alternativo* e é possível usar isso em qualquer objeto do IBM MQ.

A autoridade de usuário alternativo é essencial onde um servidor recebe solicitações de um programa e deseja assegurar que o programa possui a autoridade necessária para a solicitação. O servidor pode ter a autoridade necessária, mas precisa saber se o programa tem a autoridade para as ações solicitadas.

Por exemplo, suponha que um programa do servidor em execução com um ID do usuário PAYSERV recupere uma mensagem de solicitação de uma fila que foi colocada na fila pelo ID do usuário USER1. Quando o programa do servidor recebe a mensagem de solicitação, ele processa a solicitação e coloca a resposta de volta na fila de resposta especificada com a mensagem de solicitação. Em vez de usar seu próprio ID do usuário (PAYSERV) para autorizar a abertura da fila de resposta, o servidor pode especificar um ID do usuário diferente, neste caso, USER1. Neste exemplo, é possível usar a autoridade de usuário alternativo para controlar se PAYSERV tem permissão para especificar USER1 como um ID do usuário alternativo ao abrir a fila de resposta.

O ID do usuário alternativo é especificado no campo **AlternateUserId** do descritor de objeto.

Linux Resolvendo certos problemas de associação ao grupo no Linux

Alguns sistemas são lentos para retornar informações do grupo através da série normal de chamadas de API do sistema operacional **getgrent** e se a sua empresa tem milhares de grupos para pesquisar, procurando em quais grupos o usuário `mqm` está, a resposta lenta pode causar um tempo limite do Gerenciador de Filas interno. Para contornar esse problema, há uma API de sistema operacional alternativa.

Para usar a API alternativa, que é mais rápida, e retornar todos os grupos de uma chamada, configure a variável de ambiente MQS_GETGROUPLIST_API.

Você pode ter recebido um erro RC2035 ao conceder o acesso de conexão ao grupo secundário do usuário, e ativar a variável MQS_GETGROUPLIST_API ameniza o problema.

O IBM MQ então usa a API **getgrouplist** em vez da API **getgrent**.

Para ativar **getgrouplist**:

1. Pare o gerenciador de filas
2. Emita a exportação de comando MQS_GETGROUPLIST_API=1
3. Reinicie o gerenciador de filas

Tente novamente o cenário que falhou e, se seu problema tiver sido resolvido, você poderá considerar modificar o arquivo `.bashrc / .profile` para o usuário `mqm` para incluir essa variável de ambiente ou incluir a variável de ambiente no script usado para iniciar o gerenciador de filas.

Se o seu sistema mescla informações sobre o usuário ou o grupo para o sistema operacional por meio de vários repositórios como NIS ou LDAP, certifique-se que o ID do grupo ou do usuário seja consistente em todos os repositórios, incluindo o local, já que estes são usados para instalar e configurar permissões de nível do sistema operacional.

ALW Autoridade de contexto no AIX, Linux, and Windows

Contexto são informações que se aplicam a uma determinada mensagem e está contido no descritor de mensagens, MQMD, que faz parte da mensagem. Aplicativos podem especificar os dados de contexto quando uma chamada MQOPEN ou MQPUT é feita.

As informações de contexto são fornecidas em duas seções:

Seção de Identidade
De quem a mensagem veio. Ela consiste nos campos `UserIdentifier`, `AccountingToken` e `ApplIdentityData`.

Seção de Origem
De onde a mensagem veio, e quando ela foi colocada na fila. Ela consiste nos campos `PutApplType`, `PutApplName`, `PutDate`, `PutTime` e `ApplOriginData`.

Aplicativos podem especificar os dados de contexto quando uma chamada MQOPEN ou MQPUT é feita. Esses dados podem ser gerados pelo aplicativo, passados de outra mensagem ou gerados pelo gerenciador de filas, por padrão. Por exemplo, dados de contexto podem ser usados por programas do servidor para verificar a identidade do solicitante, testar se a mensagem veio de um aplicativo em execução com um ID do usuário autorizado.

Um programa do servidor pode usar `UserIdentifier` para determinar o ID do usuário de um usuário alternativo. Use a autorização de contexto para controlar se o usuário pode especificar qualquer uma das opções de contexto em qualquer chamada MQOPEN ou MQPUT1.

Consulte Controlando informações de contexto para obter informações sobre opções de contexto e MQMD-Descritor de mensagens para obter descrições dos campos do descritor de mensagens relacionados ao contexto.

Implementando o controle de acesso em saídas de segurança

É possível implementar o controle de acesso em uma saída de segurança usando o MCAUserIdentifier ou o gerenciador de autoridade de objeto.

MCAUserIdentifier

Cada instância de um canal atual tem uma estrutura de definição de canais associada, MQCD. Os valores iniciais dos campos no MQCD são determinados pela definição de canal que é criada por um administrador do IBM MQ. Em particular, o valor inicial de um dos campos, *MCAUserIdentifier* , é determinado pelo valor do parâmetro MCAUSER no comando DEFINE CHANNEL ou pelo equivalente a MCAUSER, se a definição de canais for criada de outra forma.

A estrutura MQCD passa para um programa de saída de canal quando é chamada por um MCA. Quando chamada, a saída de segurança pode alterar o valor de *MCAUserIdentifier*, substituindo qualquer valor especificado na definição de canais.

Multi Em Multiplataformas, a menos que o valor de *MCAUserIdentifier* esteja em branco, o gerenciador de filas usa o valor de *MCAUserIdentifier* como ID do usuário para verificações de autoridade quando um MCA tenta acessar os recursos do gerenciador de filas após ele ter se conectado ao gerenciador de filas. Se o valor de *MCAUserIdentifier* estiver em branco, o gerenciador de fila utilizará então o ID do usuário padrão do MCA. Isso se aplica aos canais RCVR, RQSTR, CLUSRCVR e SVRCONN. Para MCAs de envio, o ID do usuário padrão sempre é usado para verificações de autoridade, mesmo que o valor de *MCAUserIdentifier* não esteja em branco.

z/OS No z/OS, o gerenciador de filas pode usar o valor de *MCAUserIdentifier* para verificações de autoridade, contanto que não esteja em branco. Para MCAs de recepção e MCAs de conexão do servidor, o gerenciador de fila utiliza o valor *MCAUserIdentifier* para verificações de autoridade dependendo:

- Do valor do parâmetro PUTAUT na definição do canal
- O perfil do RACF usado para as verificações
- Do nível de acesso do ID do usuário do espaço de endereço do inicializador de canais do perfil RESLEVEL

Para MCAs de envio, depende:

- De o MCA ser um originador da chamada ou um receptor
- Do nível de acesso do ID do usuário do espaço de endereço do inicializador de canais do perfil RESLEVEL

O ID do usuário que uma saída de usuário armazena no *MCAUserIdentifier* pode ser adquirido de várias formas. Estes são alguns exemplos:

- Visto que não existe saída de segurança na extremidade do cliente de um canal MQI, um ID do usuário associado ao aplicativo do cliente IBM MQ realiza fluxo de mensagens a partir da conexão do cliente ao MCA de conexão do servidor MCA quando o aplicativo cliente emite uma chamada MQCONN.O MCA da conexão do servidor armazena este ID do usuário no campo *RemoteUserIdentifier* na estrutura de definição do canal, MQCD. Se o valor de *MCAUserIdentifier* estiver em branco nesse momento, o MCA armazenará o mesmo ID do usuário no *MCAUserIdentifier*. Se o MCA não armazenar o ID do usuário em *MCAUserIdentifier*, uma saída de segurança poderá fazê-lo posteriormente configurando *MCAUserIdentifier* com o valor de *RemoteUserIdentifier*.

  Se o ID do usuário que flui do sistema do cliente estiver entrando em um novo domínio de segurança e não for válido no sistema do servidor, a saída de segurança poderá substituir o ID do usuário por um que seja válido e armazenar o ID do usuário substituto no *MCAUserIdentifier*.
- O ID do usuário pode ser enviado pela saída de segurança do parceiro em uma mensagem de segurança.

  Em um canal de mensagens, uma saída de segurança chamada pelo MCA de envio pode enviar o ID do usuário sob o qual o MCA de envio está sendo executado. Uma saída de segurança chamada pelo MCA receptor pode então armazenar o ID do usuário no *MCAUserIdentifier* . Da mesma forma, em um canal de MQI, uma saída de segurança na extremidade do cliente do canal pode enviar o ID do usuário associado ao aplicativo do IBM MQ MQI client. Uma saída de segurança na parte do servidor do canal pode então armazenar o ID do usuário no *MCAUserIdentifier*. Como no exemplo anterior, se o ID do usuário não for válido no sistema de destino, a saída de segurança pode substituir o ID do usuário por um que seja válido e armazenar o substituto no *MCAUserIdentifier*.

  Se um certificado digital foi recebido como parte do serviço de identificação e autenticação, uma saída de segurança poderá mapear o Nome Distinto no certificado para um ID do usuário que seja válido no sistema de destino. Ele pode então armazenar o ID do usuário no *MCAUserIdentifier*.
- Se for usado TLS no canal, o Nome Distinto (DN) do parceiro será passado para a saída no campo SSLPeerNamePtr do MQCD e o DN do emissor desse certificado será passado para a saída no campo SSLRemCertIssNamePtr do MQCXP.

Para obter mais informações sobre o campo *MCAUserIdentifier*, a estrutura de definição de canal, MQCD e a estrutura de parâmetros de saída de canal, MQCXP, consulte Chamadas de saída do canal e estrutura de dados. Para obter mais informações sobre o ID do usuário que flui de um sistema do cliente em um canal MQI, consulte Controle de Acesso.

Nota: Os aplicativos de saída de segurança construídos antes da liberação do IBM WebSphere MQ 7.1 podem requerer atualização. Para obter mais informações, consulte Programas de saída de segurança de canal.

Autenticação do usuário do gerenciador de autoridade de objeto do IBM MQ

Em conexões do IBM MQ MQI client, as saídas de segurança podem ser usadas para modificar ou criar a estrutura MQCSP usada na autenticação do usuário gerenciador de autoridade de objeto (OAM). Isso é descrito em Programas de saída do canal para canais do sistema de mensagens

Implementando controle de acesso em saídas de mensagem

Talvez seja necessário usar uma saída de mensagem para substituir um ID do usuário por outro.

Considere um aplicativo cliente que envia uma mensagem para um aplicativo do servidor. O aplicativo do servidor pode extrair o ID do usuário do campo *UserIdentifier* no descritor de mensagens e, contanto que tenha autoridade de usuário alternativa, solicitar que o gerenciador de filas use esse ID do usuário para verificações de autoridade quando ele acessa recursos do IBM MQ em nome do cliente.

Se o parâmetro PUTAUT for configurado como CTX (ou ALTMCA on z/OS) na definição de canal, o ID do usuário no campo *UserIdentifier* de cada mensagem recebida será usado para verificações de autoridade quando o MCA abrir a fila de destino..

Em certas circunstâncias, quando uma mensagem de relatório é gerada, é colocada utilizando a autoridade do ID do usuário no campo *UserIdentifier* da mensagem que está causando o relatório. Em particular, os relatórios COD (confirm-on-delivery) e relatórios de expiração são sempre colocados com essa autoridade.

Em decorrência dessas situações, será necessário substituir um ID do usuário por outro no campo *UserIdentifier* à medida que uma mensagem entrar em um novo domínio de segurança. Isso pode ser feito por uma saída de mensagem na extremidade receptora do canal. Alternativamente, você pode garantir que o ID do usuário no campo *UserIdentifier* de uma mensagem de entrada seja definido no novo domínio de segurança.

Se uma mensagem de entrada contiver um certificado digital para o usuário do aplicativo que enviou a mensagem, uma saída de mensagem poderá validar o certificado e mapear o Nome Distinto no certificado para um ID do usuário que seja válido no sistema receptor. Ela poderá definir o campo *UserIdentifier* no descritor de mensagem desse ID do usuário.

Se for necessário que uma saída de mensagem altere o valor do campo *UserIdentifier* em uma mensagem de entrada, poderá ser apropriado que a saída autentique o emissor da mensagem ao mesmo tempo. Para obter mais detalhes, consulte “Mapeamento de identidade em saídas de mensagem” na página 329.

Implementando o controle de acesso na saída de API e saída cruzada da API

Uma saída da API ou saída cruzada da API pode fornecer controles de acesso para suplementar aqueles fornecidos pelo IBM MQ. Especificamente, a saída pode fornecer controle de acesso no nível de mensagem. A saída pode assegurar que um aplicativo coloque ou obtenha de uma fila somente mensagens que satisfaçam determinados critérios.

Considere os seguintes exemplos:

- Uma mensagem contém informações sobre um pedido. Quando um aplicativo tenta colocar uma mensagem em uma fila, uma saída API ou saída cruzada da API pode verificar se o valor total do pedido é menor do que algum limite prescrito.

- Mensagens chegam a uma fila de destino a partir de gerenciadores de filas remotos. Quando um aplicativo tenta obter uma mensagem da fila, uma saída API ou saída cruzada da API pode verificar se o emissor da mensagem está autorizado a enviar uma mensagem para a fila.

Multi Segurança das filas de fluxo

O recurso de filas de fluxo permite que um administrador configure uma fila local (ou modelo) com uma fila secundária, na qual as mensagens duplicadas são colocadas, sempre que uma mensagem é colocada na fila original. Há dois aspectos a serem considerados em relação às autoridades de fluxo de filas.

Autoridade para configurar uma fila para transmitir mensagens duplicadas

Para ativar o fluxo de mensagens duplicadas de uma fila para uma fila secundária, deve-se ter permissão para fazer isso. A permissão para configurar o atributo **STREAMQ** de uma fila requer que você tenha as autoridades a seguir:

1. Autoridade CHG da fila para a qual o atributo **STREAMQ** está sendo alterado
2. Autoridade CHG da fila na qual você deseja que as mensagens de duplicação sejam colocadas

A combinação dessas duas verificações de autoridade no tempo de configuração assegura que um usuário, que tenha somente a autoridade CHG na fila original, não possa colocar as mensagens em outra fila na qual ele não tem permissões.

Autoridade para abrir a fila ou filas e colocar mensagens

Quando um aplicativo abre uma fila que foi configurada com uma fila secundária, por meio de seu atributo **STREAMQ** , é feita uma verificação de autoridade que o usuário do aplicativo tem autoridade PUT na fila original.

Nota: Nenhuma verificação de autoridade adicional é feita para o usuário do aplicativo na fila secundária, que é semelhante ao modelo de autoridade usado para filas de alias.

Os aplicativos que consomem mensagens da fila original ou secundária requerem autoridade GET ou BROWSE somente na fila da qual eles estão consumindo.

Nenhuma verificação de autoridade adicional é feita no tempo de colocação ou obtenção.

exemplo

O exemplo a seguir mostra as autoridades corretas sendo configuradas para permitir que o usuário `admin` configure uma fila original, INQUIRIES.QUEUE, para transmitir suas mensagens duplicadas para a fila local ANALYTICS.queue, mas impedindo que `admin` duplique mensagens para PURCHASES.QUEUE:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(ANALYTICS.QUEUE) PRINCIPAL('admin') AUTHADD(CHG)
SET AUTHREC PROFILE(PURCHASES.QUEUE) PRINCIPAL('admin') AUTHRMV(CHG)
```

O usuário `admin` é, então, capaz de emitir o comando a seguir:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

mas se o mesmo usuário emitir o comando a seguir:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

para configurar INQUIRIES.QUEUE para colocar mensagens duplicadas em PURCHASES.QUEUE, ele receberá o erro a seguir:

```
AMQ8135E Não autorizado
```

Com INQUIRIES.QUEUE configurada para duplicar mensagens para ANALYTICS.QUEUE, os registros de autoridade a seguir são usados para permitir que um aplicativo sendo executado como usuário `appuser` coloque mensagens em INQUIRIES.QUEUE e duplique mensagens para ANALYTICS.QUEUE:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

Nota: O `appuser` não requer um registro de autoridade em ANALYTICS.QUEUE. As mensagens duplicadas são colocadas na fila pelo gerenciador de filas.

Conceitos relacionados

Filas de fluxo

z/OS Streaming queues security on z/OS

The streaming queues feature allows an administrator to configure a local (or model) queue with a secondary queue, where duplicate messages are placed, whenever a message is put to the original queue. There are two aspects to consider regarding queue streaming authorities.

Authority to configure a queue for streaming duplicate messages

If you want to enable message streaming of duplicate messages from one queue to a secondary queue, you must have permission to do so. Permission to configure the **STREAMQ** attribute of a queue requires that you have the following profiles setup:

1. ALTER access level to MQADMIN or MXADMIN for the queue they are altering the **STREAMQ** attribute for
2. ALTER access level to MQADMIN or MXADMIN for the queue you want to stream messages to

The combination of these security checks at configuration time ensures that a user, who only has ALTER access on the original queue, cannot cause messages to be put to another queue on which they have no permissions.

Authority to open the queue or queues and put messages

When an application opens a queue that has been configured with a secondary queue, through its **STREAMQ** attribute, an authority check is made that the application user has UPDATE authority on the original queue.

Note: No additional authority check is made for the application user on the secondary queue, which is similar to the authority model used for alias queues.

Applications consuming messages from either the original or the secondary queue require UPDATE or READ authority, only on the queue they are consuming from.

No additional authority checks are made at put or get time.

Example

The following example shows the correct profiles being set to allow user ADMIN to configure an original queue, INQUIRIES.QUEUE, to stream messages to local queue ANALYTICS.QUEUE using RACF:

```
RDEFINE MQCMDS <QMGR>.ALTER.QLOCAL UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.ALTER.QLOCAL CLASS(MQCMDS) ID(ADMIN) ACCESS(ALTER)

RDEFINE MQADMIN <QMGR>.QUEUE.INQUIRIES.QUEUE UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.QUEUE.INQUIRIES.QUEUE CLASS(MQADMIN) ID(ADMIN) ACCESS(ALTER)

RDEFINE MQADMIN <QMGR>.QUEUE.ANALYTICS.QUEUE UACC(NONE) OWNER(<OWNER>)
PERMIT <QMGR>.QUEUE.ANALYTICS.QUEUE CLASS(MQADMIN) ID(ADMIN) ACCESS(ALTER)
```

User ADMIN is then able to issue the following command:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

but if the same user issues the following command without setting up the correct security profiles:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

to configure INQUIRIES.QUEUE to put duplicate messages to PURCHASES.QUEUE, they receive the following error:

```
CSQM166I <QMGR> CSQMAQLC QLOCAL(INQUIRIES.QUEUE) NOT AUTHORIZED
```

Related concepts

Streaming queues

Multi Autorização LDAP

É possível usar a autorização de LDAP para remover a necessidade de um ID do usuário local.

Disponibilidade de autorização de LDAP em plataformas suportadas

A autorização do LDAP está disponível nas Multiplataformas:

Atenção:

Por meio da disponibilidade geral do IBM MQ 9.0, essa funcionalidade está disponível em todos os gerenciadores de filas novos ou migrados de uma liberação anterior.

Visão geral da autorização LDAP

Com autorização LDAP, comandos que manipulam a configuração de autorização, como **setmqaut** e **DISPLAY AUTHREC**, podem processar Nomes Distintos. Anteriormente, os usuários eram autenticados pela comparação de suas credenciais com o máximo de caracteres disponíveis que existem para usuários e grupos no sistema operacional local.

Atenção: Se você tiver executado o comando **DEFINE AUTHINFO**, deve-se reiniciar o gerenciador de filas. Se você não reiniciar o gerenciador de filas, o comando setmqaut não retornará o resultado correto.

Se um usuário fornece um ID do usuário em vez de um Nome distinto, o ID do usuário é processado. Por exemplo, quando há uma mensagem de entrada em um canal com PUTAUT(CTX), os caracteres no ID do usuário são mapeados para um Nome distinto LDAP e as verificações de autorização apropriadas são feitas.

Outros comandos como **DISPLAY CONN**, continuam a funcionar e mostram o valor real para o ID do usuário, ainda que esse ID de usuário possa não existir, de fato, no S.O. local.

Linux AIX Quando a autorização do LDAP estiver em vigor, o gerenciador de filas sempre usará o modelo de usuário de segurança em plataformas AIX and Linux, independentemente do atributo **SecurityPolicy** no arquivo qm.ini. Portanto, configurar permissões para um usuário individual afeta somente esse usuário e ninguém mais que pertença a nenhum dos grupos desse usuário.

Como com o modelo de SO, um usuário ainda tem a autoridade combinada que foi designada a ambos os individuais e para todos os grupos (se houver algum) para o qual o usuário pertence.

Por exemplo, suponha que os registros a seguir foram definidos em um repositório LDAP.

- Na classe **inetOrgPerson**:

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- Na classe **groupOfNames**:

```
dn="cn=Application Group A, ou=groups, o=yourcompany, c=yourcountry"
        longname=ApplicationGroupA [longer than 12 characters]
        members="cn=JaneDoe, ou=users, o=yourcompany, c=yourcountry",
                "cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
```

Para propósitos de autenticação, um gerenciador de filas usando esse servidor LDAP deve ter sido definido para que seu valor **CONNAUTH** aponte para um objeto **AUTHINFO** do tipo IDPWLDAP e cujos atributos de resolução do nome relevantes estão provavelmente configurados como a seguir:

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

Dada essa configuração para autenticação, um aplicativo pode concluir o campo CSPUserID, usado na chamada MQCNO, com um dos conjuntos de valores a seguir:

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

ou

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

Em qualquer caso, o sistema pode usar os valores fornecidos para autenticar o contexto do S.O. de "jodoe".

Multi

Configurando autorizações

Como você usa o nome abreviado ou **USRFIELD** para configurar autorizações.

A abordagem de trabalhar com vários formatos, descrita em “Autorização LDAP” na página 420, continua para os comandos de autorização, com uma extensão adicional que o shortname ou o USRFIELD pode ser usado em um modo sem enfeites.

A sequência de caracteres especifica um atributo específico no registro LDAP ao nomear usuários (diretores) para autorização.

Importante: A sequência de caracteres não deve conter o caractere =, porque este caractere não pode ser usado em um ID do usuário do sistema operacional.

Se você transmitir um nome do principal para o OAM para autorização que é potencialmente um shortname, a sequência de caracteres deve caber em 12 caracteres. O algoritmo de mapeamento primeiro tenta resolver a um DN usando o atributo SHORTUSR na sua consulta LDAP.

Se isso falhar com um erro UNKNOWN_ENTITY ou se a sequência especificada não puder ser uma shortname, uma tentativa adicional será feita usando o atributo USRFIELD para construir a consulta LDAP.

Atenção: Se você executou o comando DEFINE AUTHINFO, deve-se reiniciar o gerenciador de filas. Se você não reiniciar o gerenciador de filas, o comando setmqaut não retornará o resultado correto.

Para processar autorizações do usuário, as configurações do comando setmqaut a seguir são todas equivalentes.

Tabela 75. Configurações de autorização do usuário

Comando:	**Nota**
`setmqaut -m QM -t qmgr -p jodoe +connect`	Este é um nome simples, não qualificado, resolvido por meio de SHORTUSR.

Tabela 75. Configurações de autorização do usuário (continuação)	
Comando:	**Nota**
`setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect`	Além disso, um nome simples, não qualificado, resolvendo através de USRFIELD para a mesma entidade.
`setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect`	Usando um atributo denominado.
`setmqaut -m QM -t qmgr -p "phone=1234567" +connect`	Usando outro atributo denominado que não precisa ser qualquer um desses configurados no objeto AUTHINFO.

É possível usar o comando MQSC SET AUTHREC como uma alternativa para o comando **setmqaut**:

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

ou o comando Set Authority Record (MQCMD_SET_AUTH_REC) PCF com o elemento MQCACF_PRINCIPAL_ENTITY_NAMES que contém a sequência:

```
 "cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

Ao processar grupos, não há ambiguidade sobre o processamento de `shortname`, pois não há requisito para ajustar qualquer forma de um nome do grupo em 12 caracteres. Portanto, não há equivalente do atributo SHORTUSR para os grupos.

Isso significa que os exemplos de sintaxe descritos em Tabela 76 na página 422 são válidos, supondo que você tenha configurado o objeto AUTHINFO com os atributos estendidos e configurado para:

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

Tabela 76. Configurações de autorização de grupo	
Comando:	**Nota**
`setmqaut -m QM -t qmgr -g ApplicationGroupA +connect`	Usando GRPFIELD para resolver
`setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect`	Nomeando um único atributo
`setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect`	Usando o DN completo

É possível usar o comando MQSC SET AUTHREC como uma alternativa para o comando precedente **setmqaut**:

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

ou o comando Set Authority Record (MQCMD_SET_AUTH_REC) PCF com o elemento MQCACF_GROUP_ENTITY_NAMES que contém a sequência:

```
 "ApplicationGroupA"
```

Importante:

Qualquer formato que você use para fazer referência a um nome, seja para usuário ou grupo, deve ser possível derivar um DN exclusivo.

Portanto, por exemplo, não se deve ter dois registros distintos que tenham "`shortu=jodoe`".

Se um DN exclusivo único não pode ser determinado, o OAM retorna MQRC_UNKNOWN_ENTITY.

Multi

Exibindo as autorizações

Vários métodos de exibir a autorização de usuários ou grupos.

comando dspmqaut

O método mais simples para exibir as autorizações disponíveis para um usuário ou grupo é usar o comando dspmqaut.

É possível usar uma consulta em qualquer uma das variações de sintaxe para identificar um usuário ou grupo. Observe que a saída de comando repete a identidade no formato especificado na linha de comandos. A saída não relata sobre o DN completo resolvido.

Por exemplo:

```
dspmqaut -m QM -t qmgr -p johndoe
Entity johndoe has the following authorizations for object QM:
    connect
```

ou

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

comandos dmpmqaut e dmpmqcfg

O comando dmpmqaut e seus equivalentes MQSC ou PCF pode especificar o diretor ou grupo em qualquer um dos formatos suportados, como as tabelas **setmqaut** descritas em “Configurando autorizações” na página 421. No entanto, diferentemente de **dspmqaut**, o comando **dmpmqaut** sempre relata o DN completo.

```
dmpmqaut -m QM -t qmgr -p jodoe
------------------------------------
profile: self
object type:qmgr
entity:cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry
entity type: principal
authority: connect
```

Da mesma forma, o comando dmpmqcfg, que não tem nenhuma filtragem nos registros selecionados, sempre mostrará o DN completo em um formato que pode ser reproduzido posteriormente.

```
dmpmqcfg -m QM -x authrec
------------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
```

```
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

Multi Outras contraprestações ao usar a autorização LDAP

Uma breve descrição das mudanças no Message Queue Interface (MQI) e outros comandos MQSC e PCF dos quais você precisa estar ciente ao usar a autorização LDAP do IBM MQ 9.0.0.

ADOPTCTX

Não há nenhum requisito para que os aplicativos forneçam informações de autenticação ou para que o atributo ADOPTCTX seja definido como YES.

Se um aplicativo não se autenticar explicitamente ou se **ADOPTCTX** for definido como NO para o objeto CONNAUTH ativo, o contexto de identidade associado ao aplicativo será obtido do ID do usuário do sistema operacional.

Quando as autorizações precisam ser aplicadas, esse contexto é mapeado para uma identidade de LDAP usando as mesmas regras como para os comandos setmqaut.

Parâmetros de entrada para chamadas MQI

MQOPEN, MQPUT1 e MQSUB têm estruturas que permitem que um ID de usuário alternativo seja especificado.

Se esses campos são usados, o ID do usuário de 12 caracteres é mapeado para um DN usando as mesmas regras como nos comandos **setmqaut**, **dmpmqaut** e **dspmqaut**.

MQPUT e MQPUT1 também permitem que os programas adequadamente autorizados configurem o MQMD do campo UserIdentifier. O valor deste campo não é fiscalizado durante o processo PUT e pode ser definido com qualquer valor.

Como de costume, no entanto, o valor **UserIdentifier** pode ser usado para autorização em estágios posteriores do processamento de mensagens, por exemplo, quando PUTAUT(CTX) é definido em um canal de recebimento.

Nesse ponto, o identificador será verificado para obter autorização usando a configuração do gerenciador de filas de recebimento, que pode ser LDAP ou baseado no sistema operacional.

Parâmetros de saída para chamadas MQI

Sempre que um ID do usuário for fornecido para um programa em uma estrutura MQI será a versão do nome abreviado de 12 caracteres de associado à conexão.

Por exemplo, o valor de **MQAXC.UserId** para Saídas de API é o nome abreviado retornado do mapeamento do LDAP.

Outros comandos administrativos MQSC e PCF

Os comandos que mostram informações sobre o usuário no status do objeto como DISPLAY CONN USERID retornam o nome curto de 12 caracteres associado ao contexto. O DN completo não é mostrado.

Comandos que permitem que a asserção de identidades, como as regras de mapeamento CHLAUTH ou os valores de MCAUSER para os canais, podem assumir valores até o comprimento máximo definido para esses atributos (atualmente 64 caracteres).

Não há mudança para a sintaxe. Quando a autorização é necessária para essa identidade, ela é internamente mapeada para um DN usando as mesmas regras como para os comandos **setmqaut**, **dmpmqaut** e **dspmqaut**.

Isso significa que o valor MCAUSER em uma definição de canal pode não ser exibido como a mesma sequência que DISPLAY CHSTATUS, mas eles se referem à mesma identidade.

Por exemplo:

```
DEFINE CHL(SV1) CHLTYPE(SVRCONN) MCAUSER('cn=JohnDoe')
DEFINE CHL(SV2) CHLTYPE(SVRCONN) MCAUSER('jodoe')
DEFINE CHL(SV3) CHLTYPE(SVRCONN) MCAUSER('JohnDoe1@yourcompany.com')
```

Em seguida, DISPLAY CHSTATUS(*) ALL mostra o valor SHORTUSR, *MCAUSER(jodoe)* para todas as conexões.

Multi Mudando entre modelos de autorização SO e LDAP

Como alternar entre os diferentes métodos de autorização em plataformas diferentes.

O atributo CONNAUTH dos pontos de gerenciador de filas em um objeto AUTHINFO. Quando o objeto é do tipo IDPWLDAP, um repositório LDAP é usado para autenticação.

Agora é possível aplicar um método de autorização para esse mesmo objeto, o que permite continuar com a autorização baseada em S.O. ou trabalhar com autorização LDAP

IBM i, AIX and Linux

Linux IBM i AIX

O gerenciador de filas pode ser alternado a qualquer momento entre os modelos SO e LDAP. É possível mudar a configuração e fazer essa configuração ativa usando o comando REFRESH SECURITY TYPE (CONNAUTH).

Por exemplo, se esse objeto já foi configurado com as informações de conexão para autenticação:

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

Windows

Windows

Se uma mudança na configuração da autoridade envolver a comutação entre os modelos OS e LDAP, o gerenciador de filas deverá ser reiniciado para que a mudança entre em vigor. Caso contrário, será possível fazer a mudança ativa usando o comando REFRESH SECURITY TYPE (CONNAUTH).

Regras de processamento

Ao alternar do SO para autorização do LDAP, quaisquer regras de autoridade existentes do SO que foram configuradas se tornam inativas e invisíveis.

Comandos como **dmpmqaut** não exibem as regras do SO. Da mesma forma, quando comutar de volta do LDAP para SO, quaisquer autorizações LDAP definidas se tornarem inativas e invisíveis, restaurando as regras de SO originais.

Se você deseja fazer backup das definições de um gerenciador de filas por qualquer razão, usando o comando **dmpmqcfg**, então esse backup conterá apenas as regras que são definidas para o método de autorização em vigor no momento do backup.

Multi LDAP de LDAP

Uma visão geral de como cada plataforma administra o LDAP.

Ao usar autorização LDAP, a associação do grupo mqm (ou equivalente) no sistema operacional não é tão importante. Ser um membro desse grupo controla apenas se os comandos da linha de comandos determinados podem ser processados.

Em particular, deve-se estar nesse grupo para emitir os comandos strmqm e endmqm.

Quando o gerenciador de filas estiver em execução, agora existem limites na conta totalmente privilegiada. Independentemente do ID do usuário da pessoa que emite o comando **strmqm**, outros usuários pertencentes ao grupo do SO mqm (ou equivalente) não obtém privilégios especiais.

As autorizações de outros usuários são baseadas em quais grupos LDAP eles pertencem. Um uso não qualificado do nome do grupo mqm nos comandos como **setmqaut** não pode mapear para qualquer grupo LDAP.

AIX and Linux

Depois que o gerenciador de filas está em execução, a única conta com privilégios completos automáticos é o usuário real que iniciou o gerenciador de filas.

O ID do mqm ainda existe e é usado como o proprietário de recursos do sistema operacional, como arquivos, porque mqm é o ID efetivo no qual o gerenciador de filas está em execução. No entanto, o usuário mqm não será automaticamente capaz de executar tarefas administrativas controladas pelo OAM.

Windows

No Windows, as contas automaticamente privilegiadas completas é o usuário do SO que iniciou o gerenciador de filas e também o usuário que executará os processos do gerenciador de filas principal, como MUSR_MQADMIN, se o gerenciador de filas foi iniciado como um serviço do Windows.

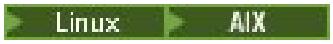
Ao executar em modo de autorização do LDAP, o Windows se comportará de forma muito semelhante às plataformas AIX and Linux. Ele lida com nomes abreviados de 12 caracteres e DNs completos.

IBM i

No IBM i, as contas automaticamente privilegiadas são as que iniciam o gerenciador de filas e o ID do QMQM.

Ambos os IDs são necessários, porque o ID do usuário que inicia o gerenciador de filas é necessário apenas para iniciar o sistema. Uma vez em execução, os processos do gerenciador de filas têm somente autoridade QMQM.

Script da amostra para fornecer privilégios MQADMIN



Como é útil ter um grupo capaz de fazer administração completa em um gerenciador de filas, um script de amostra é fornecido em plataformas AIX and Linux como:

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

Esta amostra usa dois parâmetros:

- Um nome do gerenciador de filas
- Um nome do grupo LDAP

A amostra processa os comandos setmqaut, concedendo autoridade total para todos os objetos. Este é o mesmo script que é gerado peloIBM MQ Explorer Assistente OAM para funções administrativas. Por exemplo, o código é iniciado:

```
setmqaut -t q -m qmgr -n "**" +alladm -g
      groupname
```

Confidencialidade das mensagens

As mensagens de criptografia garantem que o conteúdo das mensagens permaneça confidenciais. Há vários métodos de criptografia de mensagens no IBM MQ, dependendo de suas necessidades.

Se você precisa de proteção de dados de ponta a ponta no nível do aplicativo para sua infraestrutura do sistema de mensagens de ponto a ponto, é possível usar o Advanced Message Security para criptografar as mensagens ou escrever sua própria saída da API ou saída cruzada da API.

A solução mais segura é fornecer criptografia de ponta a ponta, criptografando uma mensagem desde o momento em que ela é emitida por um aplicativo, até o momento em que ela chega ao aplicativo de consumo. Isso pode ser feito usando o “Planejamento para o Advanced Message Security” na página 114 (AMS) ou escrevendo a própria saída de API ou saída cruzada da API. Consulte “Implementando confidencialidade em programas de saída do usuário” na página 475 para obter mais informações.

Se for necessário criptografar mensagens apenas enquanto elas estiverem sendo transportadas sobre uma rede, será possível usar o TLS. Consulte “Protocolos de segurança TLS no IBM MQ” na página 25 para obter mais informações. Como alternativa, é possível gravar a própria saída de segurança, a saída de mensagem ou enviar e receber programas de saída para executar a criptografia.

z/OS Se você precisar criptografar mensagens em repouso em um gerenciador de fila, será possível usar a criptografia do conjunto de dados do z/OS nesse gerenciador de filas; consulte “Confidentiality for data at rest on IBM MQ for z/OS with data set encryption” na página 477 para obter mais informações

Tarefas relacionadas

Conectando dois gerenciadores de filas usando TLS

Conectando um Cliente a um Gerenciador de Filas de Forma Segura

Ativando CipherSpecs

Ative um CipherSpec usando o parâmetro **SSLCIPH** no comando do MQSC **DEFINE CHANNEL** ou **ALTER CHANNEL**.

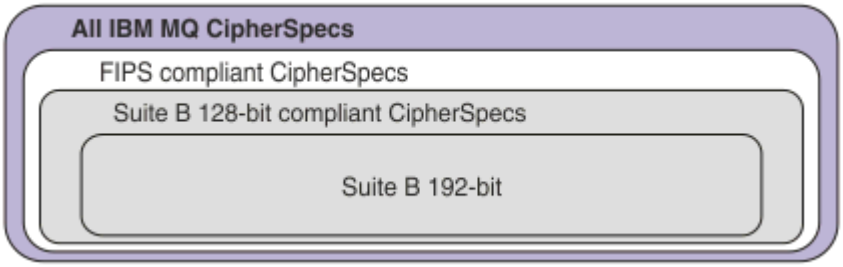
Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o IBM Crypto for C (ICC) certificado e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na NIST CMVP modules in process list.

O IBM MQ Operator 3.2.0 e a imagem do contêiner do gerenciador de filas 9.4.0.0 em diante são baseados no UBI 9 A conformidade do FIPS 140-3 está pendente atualmente e seu status pode ser visualizado procurando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" no NIST CMVP modules in process list.

Alguns dos CipherSpecs que podem ser usados com IBM MQ são compatíveis com FIPS. Alguns dos CipherSpecs compatíveis com FIPS também são compatíveis com Suite B, embora outros, como TLS_RSA_WITH_AES_256_CBC_SHA, não sejam.

Todos os CipherSpecs compatíveis com o Conjunto B também são compatíveis com FIPS. Todos os CipherSpecs compatíveis com o Suite B caem em dois grupos: 128 bit (por exemplo, ECDHE_ECDSA_AES_128_GCM_SHA256) e 192 bit (por exemplo, ECDHE_ECDSA_AES_256_GCM_SHA384),

O diagrama a seguir ilustra o relacionamento entre estes subconjuntos:



O produto suporta o protocolo de segurança TLS 1.3 em todas as plataformas..

Os CipherSpecs que podem ser usados para cada uma dessas plataformas são listados em Tabela 77 na página 428. Para obter informações sobre o uso dessas CipherSpecs, consulte “Usando o TLS 1.3 no IBM MQ” na página 432 e “IBM MQ MQI client e TLS 1.3” na página 432.

Para facilitar a configuração e a migração futura, o IBM MQ também fornece um conjunto de CipherSpecs de alias. A migração das configurações de segurança existentes para usar um CipherSpec de alias significa que é possível se adaptar a adições e descontinuações de cifras sem precisar fazer outras mudanças invasivas de configuração no futuro. Esses CipherSpecs de alias são listados na seção CipherSpecs de alias em Tabela 77 na página 428. Para obter mais informações sobre a migração para usar um CipherSpec de alias, consulte Migrando as configurações de segurança existentes para usar um CipherSpec de alias.

É possível configurar os CipherSpecs padrão conforme descrito em “Valores de CipherSpec padrão ativados no IBM MQ” na página 433. Também é possível fornecer um conjunto alternativo de CipherSpecs que estão ativados para uso com os canais em:

- Multi IBM MQ for Multiplatforms, conforme descrito no “Fornecendo uma lista customizada de CipherSpecs ordenados e ativados em IBM MQ for Multiplatforms” na página 441.
- z/OS IBM MQ for z/OS, conforme descrito no “Fornecendo uma lista customizada de CipherSpecs ordenados e ativados em IBM MQ for z/OS” na página 442.

Os CipherSpecs descontinuados que podem ser ativados novamente para serem usados com o IBM MQ, se necessário, são listados em “CipherSpecs descontinuado” na página 443.

Os CipherSpecs que você usa com o suporte do IBM MQ TLS

Os CipherSpecs que podem ser usados com o gerenciador de filas do IBM MQ automaticamente são listados na tabela a seguir. Ao exigir um certificado pessoal, você especifica um tamanho de chave para o par de chaves público e particular. O tamanho da chave que é usado durante o handshake TLS é o tamanho armazenado no certificado, a menos que ele seja determinado pelo CipherSpec, conforme indicado na tabela:

Tabela 77. CipherSpecs que podem ser usados com o suporte TLS do IBM MQ

Suporte da plataforma “1” na página 431	**Nome do CipherSpec**	**Código Hex**	**Protocolo utilizado**	**Algoritmo MAC**	**Algoritmo de criptografia (bits de criptografia)**	**FIPS** “2” na página 431	**Conjunto B**
CipherSpecs de alias							
Todos(as)	ANY_TLS13_OR_HIGHER “3” na página 431 “4” na página 431	N/D	Negociado	Negociado	Negociado	Negociado	Negociado

Tabela 77. CipherSpecs que podem ser usados com o suporte TLS do IBM MQ (continuação)

Suporte da plataforma "1" na página 431	Nome do CipherSpec	Código Hex	Protocolo utilizado	Algoritmo MAC	Algoritmo de criptografia (bits de criptografia)	FIPS "2" na página 431	Conjunto B
Todos(as)	ANY_TLS13 "4" na página 431 "5" na página 431	N/D	TLS 1.3	Negociado	Negociado	Negociado	Negociado
Todos(as)	ANY_TLS12_OR_HIGHER "4" na página 431 "6" na página 431	N/D	Negociado	Negociado	Negociado	Negociado	Negociado
Todos(as)	ANY_TLS12 "7" na página 431	N/D	TLS 1.2	Negociado	Negociado	Negociado	Negociado
Todos(as)	ANY "8" na página 431	N/D	Negociado	Negociado	Negociado	Negociado	Negociado
CipherSpecs para o TLS 1.3							
Todos(as)	TLS_AES_128_GCM_SHA256	1301	TLS 1.3	GCM	AES-128 com GCM (128)	Sim	No
Todos(as)	TLS_AES_256_GCM_SHA384	1302	TLS 1.3	GCM	AES-256 com GCM (256)	Sim	No
Todos(as)	TLS_CHACHA20_POLY1305_SHA256	1303	TLS 1.3	POLY1305	CHACHA20 (256)	No	No
ALW	TLS_AES_128_CCM_SHA256	1304	TLS 1.3	CBC-MAC	AES-128 com CTR (128)	Sim	No
ALW	TLS_AES_128_CCM_8_SHA256 "10" na página 431	1305	TLS 1.3	CBC-MAC	AES-128 com CTR (128)	Sim	No
CipherSpecs para o TLS 1.2							
Todos(as)	TLS_RSA_WITH_AES_128_CBC_SHA256 "9" na página 431	003C	TLS 1.2	SHA-256	AES (128)	Sim	No
Todos(as)	TLS_RSA_WITH_AES_256_CBC_SHA256 "9" na página 431 "11" na página 431	003D	TLS 1.2	SHA-256	AES (256)	Sim	No
Todos(as)	TLS_RSA_WITH_AES_128_GCM_SHA256 "9" na página 431 "12" na página 431	009C	TLS 1.2	SHA-256 e AEAD GCM	AES (128)	Sim	No
Todos(as)	TLS_RSA_WITH_AES_256_GCM_SHA384 "9" na página 431 "11" na página 431 "12" na página 431	009D	TLS 1.2	SHA-384 e AEAD GCM	AES (256)	Sim	No
Todos(as)	ECDHE_ECDSA_AES_128_CBC_SHA256 "9" na página 431	C023	TLS 1.2	SHA-256	AES (128)	Sim	No

Tabela 77. CipherSpecs que podem ser usados com o suporte TLS do IBM MQ (continuação)

Suporte da plataforma "1" na página 431	Nome do CipherSpec	Código Hex	Protocolo utilizado	Algoritmo MAC	Algoritmo de criptografia (bits de criptografia)	FIPS "2" na página 431	Conjunto B
Todos(as)	ECDHE_ECDSA_AES_256_CBC_SHA384 "9" na página 431 "11" na página 431	C024	TLS 1.2	SHA-384	AES (256)	Sim	No
Todos(as)	ECDHE_RSA_AES_128_CBC_SHA256 "9" na página 431	C027	TLS 1.2	SHA-256	AES (128)	Sim	No
Todos(as)	ECDHE_RSA_AES_256_CBC_SHA384 "9" na página 431 "11" na página 431	C028	TLS 1.2	SHA-384	AES (256)	Sim	No
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 "11" na página 431 "12" na página 431	C02B	TLS 1.2	SHA-256 e AEAD GCM	AES (SHA384)	Sim	128 bits
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384 "11" na página 431 "12" na página 431	C02C	TLS 1.2	SHA-384 e AEAD GCM	AES (SHA384)	Sim	192 bits
Todos(as)	ECDHE_RSA_AES_128_GCM_SHA256 "12" na página 431	C02F	TLS 1.2	SHA-256 e AEAD GCM	AES (128)	Sim	No
Todos(as)	ECDHE_RSA_AES_256_GCM_SHA384 "11" na página 431 "12" na página 431	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	Sim	No

Tabela 77. CipherSpecs que podem ser usados com o suporte TLS do IBM MQ (continuação)

Suporte da plataforma "1" na página 431	Nome do CipherSpec	Código Hex	Protocolo utilizado	Algoritmo MAC	Algoritmo de criptografia (bits de criptografia)	FIPS "2" na página 431	Conjunto B

Notas:

1. Para obter uma lista de plataformas cobertas por cada ícone de plataforma, consulte Ícones usados na documentação do produto..
2. Especifica se o CipherSpec é certificado por FIPS em uma plataforma certificada por FIPS. Consulte Federal Information Processing Standards (FIPS) para obter uma explicação do FIPS.
3. ALW A CipherSpec do alias ANY_TLS13_OR_HIGHER negocia o nível mais alto de segurança que a extremidade remota permitirá, mas se conectará apenas usando um protocolo TLS 1.3 ou superior.
4. IBM i Para usar o TLS 1.3 ou o ANY CipherSpec no IBM i, a versão do sistema operacional subjacente deve suportar o TLS 1.3. Consulte Suporte TLS do sistema para TLSv1.3 para obter mais informações.
5. ALW A CipherSpec do alias ANY_TLS13 representa um subconjunto de CipherSpecs aceitáveis que usam o protocolo TLS 1.3, conforme listado nesta tabela para cada plataforma.
6. ALW A CipherSpec do alias ANY_TLS12_OR_HIGHER negocia o nível mais alto de segurança que a extremidade remota permitirá, mas se conectará apenas usando um protocolo TLS 1.2 ou superior.
7. O CipherSpec ANY_TLS12 representa um subconjunto de CipherSpecs aceitáveis que usam o protocolo TLS 1.2, conforme listado nesta tabela para cada plataforma.
8. ALW A CipherSpec do alias ANY negocia o nível mais alto de segurança que a extremidade remota permitirá.
9. IBM i Esses CipherSpecs não são ativados em sistemas IBM i 7.4 que têm o Valor do sistema QSSLCSLCTL configurado como *OPSSYS.
10. ALW Esses CipherSpecs usam um Integrity Check Value (ICV) com 8 octetos ao invés de um ICV com 16 octetos.
11. Esse CipherSpec não pode ser usado para assegurar uma conexão a partir do IBM MQ Explorer até um gerenciador de filas a menos que os arquivos de política sem restrição sejam aplicados ao JRE usado pelo Explorer.
12. ALW Seguindo uma recomendação do GSKit, o TLS 1.2 GCM CipherSpecs tem uma restrição que significa que após 2 registros TLS24.5 serem enviados, usando a mesma chave de sessão, a conexão é finalizada com a mensagem AMQ9288E. Essa restrição do GCM está ativa, independentemente do modo FIPS que está sendo utilizado

    Para evitar que esse erro ocorra. Evite usar Cifras TLS 1.2 GCM , ative a reconfiguração de chave secreta ou inicie o gerenciador de filas ou o cliente do IBM MQ com a variável de ambiente `GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE` configurada. Para bibliotecas do GSKit , deve-se configurar essa variável de ambiente em ambos os lados da conexão e aplicá-la às conexões do cliente para o gerenciador de fila e do gerenciador de filas para as conexões do gerenciador de filas. Observe que essa configuração afeta clientes .NET não gerenciados, mas não clientes Java ou gerenciados .NET . Para obter mais informações, consulte AES-GCM restrição de cifra.

    z/OS Essa restrição não se aplica ao IBM MQ for z/OS.

Usando o TLS 1.3 no IBM MQ

O produto suporta o TLS 1.3 em todas as plataformas

Os gerenciadores de filas que foram criados no IBM MQ 9.2.0 ou mais recente suportam o TLS 1.3 por padrão. Os gerenciadores de filas migrados de versões anteriores de IBM MQ precisam ter o TLS 1.3 ativado. É possível ativar o TLS 1.3 em gerenciadores de filas migrados, configurando a propriedade **AllowTLSV13**=TRUE:

- Multi Para os gerenciadores de filas do IBM MQ for Multiplatforms, edite o arquivo `qm.ini` e inclua a propriedade **AllowTLSV13**=TRUE na sub-rotina de SSL (link para

```
SSL:
  AllowTLSV13=TRUE
```

- z/OS Para gerenciadores de filas do IBM MQ for z/OS, edite o conjunto de dados QMINI especificado na JCL de inicialização do gerenciador de filas e inclua a propriedade **AllowTLSV13**=TRUE sob a sub-rotina TransportSecurity

```
TransportSecurity:
    AllowTLSV13=TRUE
```

Quando o TLS 1.3 for ativado, e de acordo com a especificação do TLS 1.3, qualquer tentativa de comunicação com um CipherSpec fraco, independentemente de ele estar ou não ativado no IBM MQ, será rejeitada. Os CipherSpecs que o TLS 1.3 considera fracos são CipherSpecs que atendem a um ou mais dos critérios a seguir:

- Usa o protocolo SSL 3.0.
- usam RC4 ou RC2 como o algoritmo de criptografia.
- têm um tamanho de chave de criptografia (bits) igual ou menor que 112.

Essas restrições são sinalizadas com a Nota [3] na Tabela 1 de CipherSpecs descontinuados.

Se for necessário continuar usando esses CipherSpecs, o modo de TLS 1.3 deverá ser desativado:

- ALW Edite o arquivo `qm.ini` do gerenciador de filas e mude a configuração da propriedade **AllowTLSV13** para:

```
SSL:
  AllowTLSV13=FALSE
```

- z/OS Edite o conjunto de dados QMINI do gerenciador de filas e mude a configuração da propriedade **AllowTLSV13** para:

```
TransportSecurity:
    AllowTLSV13=FALSE
```

IBM MQ MQI client e TLS 1.3

ALW

Ao usar o IBM MQ MQI client, o valor de **AllowTLSV13** é inferido, a menos que seja explicitamente especificado na sub-rotina de SSL do arquivo `mqclient.ini` sendo usado pelo aplicativo.

- Se algum CipherSpecs fraco estiver ativado, **AllowTLSV13** será configurado como FALSE e nenhum CipherSpecs TLS 1.3 poderá ser usado.
- Caso contrário, **AllowTLSV13** será configurado como TRUE e os novos CipherSpecs TLS 1.3 e CipherSpecs de alias poderão ser usados.

Valores de CipherSpec padrão ativados no IBM MQ

Na configuração padrão de um novo gerenciador de filas do IBM MQ, o IBM MQ fornece suporte para os protocolos TLS 1.2 e TLS 1.3 e para vários algoritmos criptográficos usando os CipherSpecs. Para fins de compatibilidade, o IBM MQ também pode ser configurado para usar protocolos SSL 3.0 e TLS 1.0 e vários algoritmos criptográficos que são conhecidos por serem fracos ou suscetíveis a vulnerabilidades de segurança. A lista de CipherSpecs que são ativados na configuração padrão pode mudar aplicando a manutenção.

É possível configurar o IBM MQ para restringir ou permitir o uso de CipherSpecs usando os controles a seguir:

- permitir somente os CipherSpecs compatíveis com o FIPS 140-2 usando SSLFIPS.
- 

permitir somente os CipherSpecs compatíveis com o NSA Suite B usando SUITEB.
- 

Permita uma lista customizada de CipherSpecs usando **AllowedCipherSpecs**.
- 

Permita uma lista customizada de CipherSpecs usando a variável de ambiente **AMQ_ALLOWED_CIPHERS**.
- 

permitir o uso de CipherSpecs descontinuados usando **AllowWeakCipher** ou a variável de ambiente **AMQ_SSL_WEAK_CIPHER_ENABLE**.
- 

permitir o uso de CipherSpecs descontinuados usando instruções DD no CHINIT JCL.

Nota: Se você especificar uma lista customizada de CipherSpecs usando **AllowedCipherSpecs** ou **AMQ_ALLOWED_CIPHERS**, isso substituirá a ativação de quaisquer CipherSpecs descontinuados. Observe que, ao usar restrições de NSA Suite B ou FIPS 140-2 em combinação com uma lista CipherSpec customizada, deve-se assegurar-se de que a lista customizada contenha apenas CipherSpecs permitidos pelas configurações do Suite B ou FIPS 140-2.

Conceitos relacionados

"Certificados digitais e compatibilidade de CipherSpec no IBM MQ" na página 48
Este tópico fornece informações sobre como escolher os certificados digitais e do CipherSpecs para sua política de segurança, delineando o relacionamento entre os certificados digitais e de CipherSpecs no IBM MQ.

"CipherSpecs e CipherSuites" na página 22
Os protocolos de segurança criptográficos devem concordar sobre os algoritmos usados por uma conexão segura. CipherSpecs e CipherSuites definem combinações específicas de algoritmos.

"Configurando o IBM MQ para o Conjunto B" na página 45
O IBM MQ pode ser configurado para operar em conformidade com o padrão do Conjunto B do NSA em plataformas AIX, Linux, and Windows.

"FIPS (Federal Information Processing Standards)" na página 35
Este tópico apresenta o Federal Information Processing Standards (FIPS) Cryptomodule Validation Program do National Institute of Standards and Technology dos EUA e as funções criptográficas que podem ser usadas nos canais TLS.

Tarefas relacionadas

Migrando as configurações de segurança existentes para usar um CipherSpec de alias

Referências relacionadas

DEFINE CHANNEL

ALTER CHANNEL

Alterar, Copiar e Criar Canal

ALW AES-restrição de cifra doGCM

Um guia para as restrições que são impostas às cifras AES-GCM quando usadas para a criptografia TLS. Essas restrições são impostas pelas organizações IETF e NIST e requerem que a mesma chave de sessão não seja usada para transferir com segurança mais de 2 registros TLS[24.5] ao usar cifras AES-GCM .

Para obter mais informações sobre essas restrições, consulte a Seção RFC 9325 4.4 Limites no Uso da Chave e Seção RFC 8446 5.5

O IBM MQ não implementa a funcionalidade criptográfica diretamente Em vez disso, várias bibliotecas criptográficas diferentes são usadas para fornecer a funcionalidade TLS e Advanced Message Security . Nos sistemas operacionais Windows, Linuxe AIX , a biblioteca criptográfica que o IBM MQ usa é IBM Global Security Kit (GSKit) Para aplicativos, as bibliotecas C e .NET não gerenciadas usam GSKit para funcionalidade criptográfica. A implementação dos algoritmos de criptografia AES-GCM por GSKit inclui as restrições especificadas pelo grupo de padrões. Além disso, essas restrições são ativadas por padrão.. Dessa forma, a IBM MQ comunicação TLS, ao usar cifras AES-GCM , será finalizada se mais de 2 registros TLS[24.5] forem transmitidos usando a mesma chave de sessão.

Nota: Essa restrição não está presente nas plataformas IBM i, IBM Z ou IBM MQ for HPE NonStop ou aplicativos Java/JMSgerenciados .NET porque bibliotecas criptográficas diferentes são usadas e essas bibliotecas não implementaram a mesma restrição.

Se um canal IBM MQ permanecer em execução por tempo suficiente para que mais de 2 registros TLS[24.5] sejam transmitidos usando a mesma chave de sessão, a biblioteca criptográfica subjacente terminará a conexão. Isso faz o canal ser finalizado e uma mensagem de erro AMQ9288E é gerada. Os aplicativos que tiverem sua comunicação finalizada dessa maneira receberão um código de retorno `MQRC_CONNECTION_BROKEN` de qualquer operação IBM MQ que estiver sendo executada

A finalização da conexão pode ser executada em qualquer extremidade da comunicação, mas apenas em extremidades que estão usando GSKit para a funcionalidade criptográfica.

Conselhos para atenuar a restrição

Algumas opções para como evitar ou manipular comunicações que são finalizadas devido a essa restrição são as seguintes::

Usar clientes reconectáveis

Os aplicativos podem ser configurados para tentar automaticamente uma reconexão, se uma conexão falhar. Isso inclui conexões que são finalizadas devido à restrição do GCM Quando configurado para reconexão, o aplicativo cliente é restaurado automaticamente em qualquer ponto de falha e quaisquer identificadores para objetos abertos são restaurados Isso é feito sem retornar para o código do aplicativo.

Para obter mais informações, consulte Reconexão automática do cliente.

Configure um valor de reconfiguração de chave secreta

IBM MQ pode ser configurado para solicitar uma reconfiguração de chave de sessão após um número configurável de bytes ter sido transferido através de um canal Ao atingir esse limite, o IBM MQ solicita que a camada de criptografia execute uma reconfiguração de chave de sessão, resultando em uma nova chave de sessão.

É importante observar que o valor especificado é o número de bytes transferidos, que está relacionado ao tamanho das mensagens enviadas pelo IBM MQ. A restrição está no número de registros TLS enviados. Não há um mapeamento direto entre bytes de mensagens e registros TLS, pois um registro TLS pode enviar um número máximo de bytes dependentes da Maximum Transmission Unit (MTU) da rede. Quaisquer mensagens enviadas maiores que esse valor são transmitidas como diversos registros TLS. O valor de MTU varia entre as redes. Além disso, há outras razões pelas quais um registro TLS pode precisar ser enviado fora da transmissão de dados de mensagem do IBM MQ , por exemplo, IBM MQ Verificações de pulsação, alertas TLS, outras mensagens de protocolo IBM MQ Esses registros TLS adicionais contam para o número máximo de registros TLS, mas não são contados no valor de reconfiguração de chave secreta IBM MQ .

Reconfigurar regularmente uma chave de sessão usando a reconfiguração de chave secreta pode evitar que o canal seja finalizado devido à restrição AES-GCM .

Para obter mais informações, consulte Reconfigurando chaves secretas SSL e TLS.

Usar especificações de código TLS 1.3

Embora a restrição AES-GCM ainda esteja presente ao usar o protocolo TLS 1.3 , o protocolo TLS 1.3 suporta a execução automática de uma reconfiguração de chave de sessão sem a necessidade de interromper as comunicações TLS Isso permite que o GSKit gerencie a reconfiguração da chave de sessão quando for necessário sem IBM MQ precisar solicitar uma reconfiguração de chave secreta.

Para obter mais informações, consulte Usando o TLS 1.3 em IBM MQ em "Ativando CipherSpecs" na página 427

Desativar a restrição AES-GCM

Se necessário, a restrição poderá ser desativada configurando a variável de ambiente **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** para desativar a restrição AES-GCM . Isso permite que qualquer número de registros TLS seja enviado usando a mesma chave de sessão. Se escolher essa mitigação, a variável de ambiente deverá ser configurada em cada extremidade de comunicação que usa GSKit para comunicações seguras.

Aviso: Essa opção não é recomendada porque, após mais de 2 registros TLS$^{24.5}$ terem sido enviados, é possível que os invasores executem a análise nos registros enviados para determinar a chave de sessão em uso. Quando a chave de sessão tiver sido determinada, toda a comunicação existente e futura usando essa chave de sessão será comprometida.

Ordem do CipherSpec no handshake TLS

A ordem de CipherSpecs é usada ao escolher entre diversos CipherSpecs possíveis, por exemplo, ao usar um dos CipherSpecs ANY*.

Durante um handshake TLS, um cliente e um servidor trocam os CipherSpecs e os protocolos que eles dão suporte na ordem em que preferirem. Um CipherSpec comum que ambos os lados priorizam é escolhido e usado para a comunicação TLS. Ao escolher um protocolo de CipherSpec, a versão também será considerada. Por exemplo, se um servidor listar os CipherSpecs do TLS 1.2 antes dos CipherSpecs do TLS 1.3, ele ainda priorizará o TLS 1.3, desde que ele seja suportado pelo cliente e tenha um CipherSpec do TLS 1.3 comum que possa ser usado.

Quando o IBM MQ é configurado para TLS, ele configura o CipherSpecs na ordem mostrada na tabela a seguir, da mais preferencial para a menos preferencial

Nota: Se um CipherSpec não for ativado por meio do atributo **AllowedCipherSpecs**, ele não será configurado para uso durante um handshake TLS.

Caso o atributo **AllowedCipherSpecs** não seja especificado, uma lista padrão de cifras ativadas, indicada pela tabela a seguir, será usada.

Tabela 78. Ordem dos CipherSpecs do IBM MQ 9.2.0

Plataforma	CipherSpec	Protocolo	Código Hexadecimal	Ativado por padrão
Todos(as)	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sim
Todos(as)	TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sim
Todos(as)	TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sim
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	1304	Sim

Tabela 78. Ordem dos CipherSpecs do IBM MQ 9.2.0 (continuação)				
Plataforma	**CipherSpec**	**Protocolo**	**Código Hexadecimal**	**Ativado por padrão**
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	1305	Sim
Todos(as)	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sim
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	C02C	Sim
Todos(as)	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sim
Todos(as)	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sim
Todos(as)	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sim
Todos(as)	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sim
Todos(as)	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sim
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	C02B	Sim
Todos(as)	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sim
Todos(as)	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sim
Todos(as)	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sim
Todos(as)	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sim
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	C008	No
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	C012	No
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	0005	No

Tabela 78. Ordem dos CipherSpecs do IBM MQ 9.2.0 (continuação)

Plataforma	CipherSpec	Protocolo	Código Hexadecimal	Ativado por padrão
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	C007	No
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	C011	No
Todos(as)	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	No
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	C006	No
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	C010	No
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	0000	No
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	No
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	No
IBM i	AES_SHA_US	TLS 1.0	002E	No
Todos(as)	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	No
Todos(as)	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	No
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	0004	No
Todos(as)	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	No
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	0003	No
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	0006	No
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	0002	No
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	0001	No
Todos(as)	TRIPLE_DES_SHA_US	SSL v3	000A	No

Tabela 78. Ordem dos CipherSpecs do IBM MQ 9.2.0 (continuação)				
Plataforma	**CipherSpec**	**Protocolo**	**Código Hexadecimal**	**Ativado por padrão**
Todos(as)	RC4_SHA_US	SSL v3	0005	No
Todos(as)	RC4_MD5_US	SSL v3	0004	No
Todos(as)	DES_SHA_EXPORT	SSL v3	0009	No
Todos(as)	RC4_MD5_EXPORT	SSL v3	0003	No
Todos(as)	RC2_MD5_EXPORT	SSL v3	0006	No
Todos(as)	NULL_SHA	SSL v3	0002	No
Todos(as)	NULL_MD5	SSL v3	0001	No
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	FEFF	No
ALW	RC4_56_SHA_EXPORT1024	SSL v3	0064	No
ALW	DES_SHA_EXPORT1024	SSL v3	0062	No
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	FEFE	No

Esta lista foi construída ordenando os protocolos com a lista padrão fornecida pela biblioteca criptográfica usada por IBM MQ em z/OS e é consistente em todas as plataformas z/OS e distribuídas.

Alterando a Ordem

Se uma ordem diferente é desejada, então uma nova ordem de CipherSpecs pode ser fornecida usando o atributo **AllowedCipherSpecs** da sub-rotina SSL em IBM MQ for Multiplatforms z/OS ou a sub-rotina TransportSecurity em IBM MQ for z/OS, com as regras a seguir:

- As versões de protocolo mais altas são sempre usadas, independentemente da posição delas na lista.
- Todos os CipherSpecs desativados serão ativados novamente se forem fornecidos na lista.
- A ordem de lista do servidor TLS tem uma prioridade mais alta do que o cliente TLS
- Quando o TLS 1.3 for ativado, determinados CipherSpecs não serão suportados.

Por exemplo, no IBM MQ for Multiplatforms, se o item a seguir estiver configurado no gerenciador de filas:

```
SSL:
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,
TLS_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

z/OS e no IBM MQ for z/OS, se o item a seguir estiver configurado no gerenciador de filas:

```
TransportSecurity:
AllowedCipherSpecs=TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_128_GCM_SHA256,
TLS_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA
```

em seguida:

- Um cliente que se conecta com o ANY_TLS12 provavelmente usará o CipherSpec do TLS 1.2 TLS_RSA_WITH_AES_128_GCM_SHA256.
- Um cliente que se conecta com o ANY_TLS12_OR_HIGHER provavelmente usará o CipherSpec do TLS 1.3 TLS_AES_128_GCM_SHA256 (supondo que o cliente suporta o TLS 1.3).

- Um cliente que se conecta com o CipherSpec do TLS 1.0 TLS_RSA_WITH_AES_256_CBC_SHA usará esse CipherSpec.

Versões anteriores do IBM MQ

Antes de IBM MQ 9.2.0, a ordem a seguir de CipherSpecs foi usada:

Tabela 79. A ordem dos CipherSpecs antes do IBM MQ 9.2.0

Plataforma	CipherSpec	Protocolo	Ativado por padrão
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	No
IBM i	AES_SHA_US	TLS 1.0	No
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	No
Todos(as)	RC4_SHA_US	SSL v3	No
Todos(as)	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	No
Todos(as)	RC4_MD5_US	SSL v3	No
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	No
Todos(as)	TRIPLE_DES_SHA_US	SSL v3	No
Todos(as)	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	No
ALW	DES_SHA_EXPORT1024	SSL v3	No
Todos(as)	RC4_56_SHA_EXPORT1024	SSL v3	No
Todos(as)	RC4_MD5_EXPORT	SSL v3	No
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	No
Todos(as)	RC2_MD5_EXPORT	SSL v3	No
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	No
Todos(as)	DES_SHA_EXPORT	SSL v3	No
Todos(as)	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	No
Todos(as)	NULL_SHA	SSL v3	No
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	No
Todos(as)	NULL_MD5	SSL v3	No

Tabela 79. A ordem dos CipherSpecs antes do IBM MQ 9.2.0 (continuação)			
Plataforma	**CipherSpec**	**Protocolo**	**Ativado por padrão**
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	No
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	No
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	No
Todos(as)	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	Sim
Todos(as)	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	Sim
Todos(as)	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	No
Todos(as)	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	Sim
Todos(as)	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	Sim
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	No
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	No
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	No
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	No
Todos(as)	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	Sim
Todos(as)	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	Sim
Todos(as)	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	Sim
Todos(as)	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	Sim
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	Sim
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	Sim
Todos(as)	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	Sim
Todos(as)	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	Sim
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	No

Tabela 79. A ordem dos CipherSpecs antes do IBM MQ 9.2.0 (continuação)			
Plataforma	**CipherSpec**	**Protocolo**	**Ativado por padrão**
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	No
ALW	TLS_RSA_WITH_NULL_NULL	TLS 1.2	No
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	No
Multi	TLS_AES_128_GCM_SHA256	TLS 1.3	Sim
Multi	TLS_AES_256_GCM_SHA384	TLS 1.3	Sim
Multi	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	Sim
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	Sim
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	Sim

Importante: A partir de 23 de julho de 2020, o atributo AllowedCipherSpecs a seguir ativa apenas os CipherSpecs que estão ativados atualmente por padrão. No entanto, é necessário verificar os CipherSpecs ativados pelo atributo AllowedCipherSpecs a seguir com dados atuais, para assegurar que os CipherSpecs que foram descontinuados desde essa data não sejam ativados novamente inadvertidamente.

Se for necessário retornar a essa ordem dos CipherSpecs, será possível fazê-lo usando o valor de atributo de sub-rotina SSL/TransportSecurity do **AllowedCipherSpecs** a seguir:

```
AllowedCipherSpecs=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,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
```

Fornecendo uma lista customizada de CipherSpecs ordenados e ativados em IBM MQ for Multiplatforms

Multi

É possível fornecer um conjunto alternativo de CipherSpecs que estão ativados e, em sua ordem de preferência, para uso com canais IBM MQ , usando a ALW **AMQ_ALLOWED_CIPHERS** variável de ambiente ou o atributo de sub-rotina SSL **AllowedCipherSpecs** do arquivo `.ini` . Talvez você queira usar essa configuração por uma das seguintes razões:

- Para restringir os listeners do IBM MQ de aceitarem solicitações de início de canal de entrada, a menos que eles usem um dos CipherSpecs nomeados.
- Para mudar a ordem de prioridade dos CipherSpecs que são usados em um handshake TLS.

Essa funcionalidade pode ser usada para controlar os CipherSpecs que estão incluídos no ANY* CipherSpecs.

A variável de ambiente **AMQ_ALLOWED_CIPHERS** ou o atributo de sub-rotina SSL **AllowedCipherSpecs** aceita:

- Um único nome CipherSpec.

- Uma lista separada por vírgula dos nomes de CipherSpec para ativar novamente.
- O valor especial de ALL, representando todas os CipherSpecs.

Nota: Não se deve ativar **ALL** os CipherSpecs, pois isso ativará os protocolos SSL 3.0 e TLS 1.0 e um grande número de algoritmos criptográficos fracos.

Se essa configuração estiver definida, ela substituirá a lista de CipherSpec padrão e fará com que o IBM MQ ignore as configurações de descontinuação de cifra fraca (veja abaixo):

- Os listeners do IBM MQ só aceitam propostas SSL/TLS que usam um dos CipherSpecs nomeados.
- Os canais do IBM MQ só permitem um valor SSLCIPH em branco ou um dos CipherSpecs nomeados.
- A conclusão dos valores de SSLCIPH da guia **runmqsc** restringe os valores de conclusão a um dos CipherSpecs nomeados.

Por exemplo, se você quer apenas permitir que canais sejam definidos/alterados e que listeners aceitem ECDHE_RSA_AES_128_GCM_SHA256 ou ECDHE_ECDSA_AES_256_GCM_SHA384, é possível configurar o seguinte no arquivo qm.ini:

```
SSL:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

Além disso, os CipherSpecs nessa lista serão usados para determinar a prioridade dos CipherSpecs usados durante um handshake TLS. Por exemplo, se você especificar uma lista de TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256, será provável que, durante o handshake, se um cliente se conectar especificando ambos esses CipherSpecs, ou seja, com o ANY_TLS12, o CipherSpec TLS_RSA_WITH_AES_128_CBC_SHA256 será escolhido em vez do CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256.

Observe que as cifras usadas pelos canais AMQP ou MQTT podem ser restritas usando as configurações de arquivo java.security.

Fornecendo uma lista customizada de CipherSpecs ordenados e ativados em IBM MQ for z/OS

É possível que você forneça um conjunto alternativo de CipherSpecs ativados e em sua ordem de preferência para uso com os canais do IBM MQ, usando o atributo de sub-rotina do TransportSecurity **AllowedCipherSpecs** do Conjunto de dados do QMINI. É possível que você queira fazer isso por qualquer um dos motivos a seguir:

- Para restringir os listeners do IBM MQ de aceitarem solicitações de início de canal de entrada, a menos que eles usem um dos CipherSpecs nomeados.
- Para mudar a ordem de prioridade dos CipherSpecs que são usados em um handshake TLS.

É possível usar essa funcionalidade para controlar os CipherSpecs que estão incluídos nos CipherSpecs ANY*. O atributo **AllowedCipherSpecs** aceita:

- Um único nome CipherSpec.
- Uma lista separada por vírgula dos nomes de CipherSpec para ativar novamente.
- O valor especial de ALL, representando todas os CipherSpecs.

  Nota: Não se deve ativar **ALL** os CipherSpecs, pois isso ativará os protocolos SSL 3.0 e TLS 1.0 e um grande número de algoritmos criptográficos fracos. Se você definir essa configuração, ela substituirá a lista de CipherSpec padrão e fará com que o IBM MQ ignore as configurações de descontinuação de cifras fracas. Consulte “Ativando CipherSpecs descontinuados no z/OS” na página 447.

Os listeners de IBM MQ só aceitam propostas de SSL/TLS que usam um dos CipherSpecs nomeados e canais do IBM MQ que permitem apenas um valor SSLCIPH em branco ou um dos CipherSpecs nomeados.

Por exemplo, para permitir apenas que os canais sejam definidos/alterados e que os listeners aceitem ECDHE_RSA_AES_128_GCM_SHA256 ou ECDHE_RSA_AES_256_GCM_SHA384, é possível configurar o item a seguir:

```
TransportSecurity:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                     ECDHE_RSA_AES_256_GCM_SHA384
```

Além disso, os CipherSpecs nessa lista são usados para determinar a prioridade dos CipherSpecs usados durante um handshake TLS. Por exemplo, se você especificar uma lista de TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256, será provável que, durante o handshake, se um cliente se conectar especificando ambos esses CipherSpecs, ou seja, com o ANY_TLS12, o CipherSpec TLS_RSA_WITH_AES_128_CBC_SHA256 será escolhido em vez do CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256.

Deprecated CipherSpecs descontinuado

Uma lista de CipherSpecs descontinuados que é possível usar com o IBM MQ, se necessário.

Os CipherSpecs descontinuados que podem ser usados com suporte de TLS do IBM MQ são listados na tabela a seguir.

Tabela 80. CipherSpecs descontinuados que podem ser reativados para uso com o IBM MQ

Suporte da plataforma “1” na página 446	**Nome do CipherSpec**	**Código Hex**	**Protocolo utilizado**	**Integridade de dados**	**Algoritmo de criptografia (bits de criptografia)**	**FIPS** “2” na página 446	**Conjunto B**	**Atualizar quando descontinuado**
CipherSpecs para o SSL 3.0								
IBM i	AES_SHA_US “3” na página 446	002F	SSL 3.0	SHA-1	AES (128)	No	No	9.0.0.0
Todos(as)	DES_SHA_EXPORT “3” na página 446 “4” na página 446 “5” na página 446	0009	SSL 3.0	SHA-1	DES (56)	No	No	9.0.0.0
ALW	DES_SHA_EXPORT1024 “3” na página 446 “6” na página 446	0062	SSL 3.0	SHA-1	DES (56)	No	No	9.0.0.0
ALW	FIPS_WITH_DES_CBC_SHA “3” na página 446	FEFE	SSL 3.0	SHA-1	DES (56)	Não “7” na página 446	No	9.0.0.0
ALW	FIPS_WITH_3DES_EDE_CBC_SHA “3” na página 446	FEFF	SSL 3.0	SHA-1	3DES (168)	Não “8” na página 446	No	9.0.0.1 e 9.0.1
Todos(as)	NULL_MD5 “3” na página 446	0001	SSL 3.0	MD5	Nenhum	No	No	9.0.0.1
Todos(as)	NULL_SHA “3” na página 446	0002	SSL 3.0	SHA-1	Nenhum	No	No	9.0.0.1
Todos(as)	RC2_MD5_EXPORT “3” na página 446 “4” na página 446 “5” na página 446	0006	SSL 3.0	MD5	RC2 (40)	No	No	9.0.0.0

Tabela 80. CipherSpecs descontinuados que podem ser reativados para uso com o IBM MQ (continuação)

Suporte da plataforma "1" na página 446	Nome do CipherSpec	Código Hex	Protocolo utilizado	Integridade de dados	Algoritmo de criptografia (bits de criptografia)	FIPS "2" na página 446	Conjunto B	Atualizar quando descontinuado
Todos(as)	RC4_MD5_EXPORT "4" na página 446 "3" na página 446	0003	SSL 3.0	MD5	RC4 (40)	No	No	9.0.0.0
Todos(as)	RC4_MD5_US "3" na página 446	0004	SSL 3.0	MD5	RC4 (128).	No	No	9.0.0.0
Todos(as)	RC4_SHA_US "3" na página 446 "5" na página 446	0005	SSL 3.0	SHA-1	RC4 (128).	No	No	9.0.0.0
ALW	RC4_56_SHA_EXPORT1024 "3" na página 446 "6" na página 446	0064	SSL 3.0	SHA-1	RC4 (56)	No	No	9.0.0.0
Todos(as)	TRIPLE_DES_SHA_US "3" na página 446 "5" na página 446	000A	SSL 3.0	SHA-1	3DES (168)	No	No	9.0.0.1 e 9.0.1
CipherSpecs para o TLS 1.0								
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5 "3" na página 446	0006	TLS 1.0	MD5	RC2 (40)	No	No	9.0.0.0
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5 "3" na página 446 "4" na página 446	0003	TLS 1.0	MD5	RC4 (40)	No	No	9.0.0.0
Todos(as)	TLS_RSA_WITH_DES_CBC_SHA "3" na página 446	0009	TLS 1.0	SHA-1	DES (56)	Não "9" na página 446	No	9.0.0.0
IBM i	TLS_RSA_WITH_NULL_MD5 "3" na página 446	0001	TLS 1.0	MD5	Nenhum	No	No	9.0.0.1
IBM i	TLS_RSA_WITH_NULL_SHA "3" na página 446	0002	TLS 1.0	SHA-1	Nenhum	No	No	9.0.0.1
IBM i	TLS_RSA_WITH_RC4_128_MD5 "3" na página 446	0004	TLS 1.0	MD5	RC4 (128).	No	No	9.0.0.0
z/OS ALW	TLS_RSA_WITH_AES_128_CBC_SHA "10" na página 446	002F	TLS 1.0	SHA-1	AES (128)	Sim	No	9.0.5
z/OS ALW	TLS_RSA_WITH_AES_256_CBC_SHA "6" na página 446 "10" na página 446	0035	TLS 1.0	SHA-1	AES (256)	Sim	No	9.0.5
Todos(as)	TLS_RSA_WITH_3DES_EDE_CBC_SHA	000A	TLS 1.0	SHA-1	3DES (168)	Sim	No	9.0.0.1 e 9.0.1
CipherSpecs para o TLS 1.2								
ALW	ECDHE_ECDSA_NULL_SHA256 "3" na página 446	C006	TLS 1.2	SHA-1	Nenhum	No	No	9.0.0.1
ALW	ECDHE_ECDSA_RC4_128_SHA256 "3" na página 446	C007	TLS 1.2	SHA-1	RC4 (128).	No	No	9.0.0.0

Tabela 80. CipherSpecs descontinuados que podem ser reativados para uso com o IBM MQ (continuação)

Suporte da plataforma "1" na página 446	Nome do CipherSpec	Código Hex	Protocolo utilizado	Integridade de dados	Algoritmo de criptografia (bits de criptografia)	FIPS "2" na página 446	Conjunto B	Atualizar quando descontinuado
ALW IBM i	ECDHE_RSA_NULL_SHA256 "3" na página 446	C010	TLS 1.2	SHA-1	Nenhum	No	No	9.0.0.1
ALW IBM i	ECDHE_RSA_RC4_128_SHA256 "3" na página 446	C011	TLS 1.2	SHA-1	RC4 (128).	No	No	9.0.0.0
ALW	TLS_RSA_WITH_NULL_NULL "3" na página 446	0000	TLS 1.2	Nenhum	Nenhum	No	No	9.0.0.1
Todos(as)	TLS_RSA_WITH_NULL_SHA256 "3" na página 446	003B	TLS 1.2	SHA-256	Nenhum	No	No	9.0.0.1
ALW	TLS_RSA_WITH_RC4_128_SHA256 "3" na página 446	0005	TLS 1.2	SHA-1	RC4 (128).	No	No	9.0.0.0
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	C0008	TLS 1.2	SHA-1	3DES (168)	Sim	No	9.0.0.1 e 9.0.1
ALW IBM i	ECDHE_RSA_3DES_EDE_CBC_SHA256	C012	TLS 1.2	SHA-1	3DES (168)	Sim	No	9.0.0.1 e 9.0.1

Tabela 80. CipherSpecs descontinuados que podem ser reativados para uso com o IBM MQ (continuação)

Suporte da plataforma "1" na página 446	Nome do CipherSpec	Código Hex	Protocolo utilizado	Integridade de dados	Algoritmo de criptografia (bits de criptografia)	FIPS "2" na página 446	Conjunto B	Atualizar quando descontinuado

Notas:

1. Para obter uma lista de plataformas cobertas por cada ícone de plataforma, consulte Ícones usados na documentação do produto..
2. Especifica se o CipherSpec é certificado por FIPS em uma plataforma certificada por FIPS. Consulte Federal Information Processing Standards (FIPS) para obter uma explicação do FIPS.
3. ALW Esses CipherSpecs são desativados quando o TLS 1.3 é ativado (por meio da propriedade AllowTLSV13 no `qm.ini`).

   z/OS Os gerenciadores de filas criados no IBM MQ for z/OS 9.2.0 ou mais recente ativam o TLS 1.3 por padrão, o que desativa esses CipherSpecs. Será possível ativar esses CipherSpecs, se necessário, desligando o TLS V1.3. Isso é feito incluindo **AllowTLSV13**=*FALSE* à sub-rotina TransportSecurity do conjunto de dados QMINI no gerenciador de filas JCL. Os gerenciadores de filas migrados para a IBM MQ for z/OS 9.2.0 de uma versão anterior não têm o TLS 1.3 ativado por padrão e, portanto, têm esses CipherSpecs ativados.
4. O tamanho de chave de handshake máximo é 512 bits. Caso nenhum dos certificados trocados durante o protocolo de reconhecimento do SSL tenha um tamanho de chave superior a 512 bits, uma chave de 512 bits temporária será gerada para uso durante o protocolo de reconhecimento.
5. Esses CipherSpecs não são mais suportados pelo IBM MQ classes for Java ou IBM MQ classes for JMS. Para obter mais informações, consulte SSL/TLS CipherSpecs e CipherSuites em IBM MQ classes for Java ou SSL/TLS CipherSpecs e CipherSuites em IBM MQ classes for JMS.
6. O tamanho de chave de handshake é 1024 bits.
7. Deprecated Este CipherSpec era certificado FIPS 140-2 antes de 19 de Maio de 2007. O nome do `FIPS_WITH_DES_CBC_SHA` é histórico e reflete o fato de que esse CipherSpec era anteriormente (mas não é mais) compatível com FIPS. Esse CipherSpec foi descontinuado e seu uso não é recomendado.
8. Deprecated O nome do `FIPS_WITH_3DES_EDE_CBC_SHA` é histórico e reflete o fato de que esse CipherSpec era anteriormente (mas não é mais) compatível com FIPS. O uso deste CipherSpec está descontinuado.
9. Este CipherSpec era certificado FIPS 140-2 antes de 19 de Maio de 2007.
10. Reativar apenas esses CipherSpecs não requer o uso da instrução CSQXWEAK DD.

Ativando CipherSpecs descontinuados no IBM MQ for Multiplatforms

Multi

Por padrão, você não tem permissão para especificar um CipherSpec descontinuado em uma definição de canal. Se você tentar especificar um CipherSpec descontinuado em IBM MQ for Multiplatforms, receberá a mensagem AMQ8242: definição de SSLCIPH incorreta e PCF retorna `MQRCCF_SSL_CIPHER_SPEC_ERROR`.

Não é possível iniciar um canal com um CipherSpec descontinuado. Se você tentar fazer isso com um CipherSpec descontinuado, o sistema retornará `MQCC_FAILED (2)`, juntamente com um **Reason** de `MQRC_SSL_INITIALIZATION_ERROR (2393)` para o cliente.

É possível re-ativar uma ou mais das CipherSpecs descontinuadas para definição de canais, no tempo de execução no servidor, configurando a variável de ambiente **AMQ_SSL_WEAK_CIPHER_ENABLE**.

A variável de ambiente **AMQ_SSL_WEAK_CIPHER_ENABLE** aceita:

- Um nome de CipherSpec único ou
- Uma lista separada por vírgula de nomes de CipherSpec para ativar novamente ou
- O valor especial de ALL, representando todas os CipherSpecs.

Atenção: Embora ALL seja uma opção válida, use-a **apenas** quando especificamente necessário para sua empresa, pois ativar novamente ALL CipherSpecs ativa os protocolos SSL 3.0 e TLS 1.0, bem como um grande número de algoritmos criptográficos fracos.

Por exemplo, se você desejar ativar novamente ECDHE_RSA_RC4_128_SHA256, configure a seguinte variável de ambiente:

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

ou, alternativamente mude a sub-rotina SSL no arquivo `qm.ini` configurando:

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

Ativando CipherSpecs descontinuados no z/OS

z/OS

Por padrão, você não tem permissão para especificar um CipherSpec descontinuado em uma definição de canal. Se você tentar especificar um CipherSpec descontinuado em z/OS, receberá uma mensagem CSQM102E, CSQX616E ou CSQX674E.

Siga as instruções listadas nesta seção se você receber qualquer uma dessas mensagens e sua empresa precisar reativar o uso de CipherSpecs fracos.

Atenção: Nas instruções a seguir, para que as instruções de definição simulada (DD) entrem em vigor, SSLTASKS deve ser um valor diferente de zero. Se isso exigir uma mudança para SSLTASKS, deve-se reciclar o iniciador de canais.

No IBM MQ for z/OS, o método atual de controle de CipherSpecs fracos ou quebrados é o seguinte:

- Se você quiser reativar o uso de CipherSpecs fracos, faça isso incluindo uma instrução de definição de dados (DD) fictícia chamada CSQXWEAK na JCL do inicializador de canais. Se especificada por conta própria, isso ativa apenas CipherSpecs fracos associados ao protocolo TLS 1.2, por exemplo:

  ```
  //CSQXWEAK DD DUMMY
  ```

  Nota: Nem todos os CipherSpecs obsoletos exigem o uso dessa instrução DD, consulte a observação 10 na tabela anterior.
- Se você deseja reativar o uso de SSLv3 CipherSpecs, faça isso incluindo também uma instrução DD simulada denominada CSQXSSL3 na JCL do iniciador de canais. Todos os SSLv3 CipherSpecs são considerados **fracos**, portanto, também deve-se especificar CSQXWEAK:

  ```
  //CSQXSSL3 DD DUMMY
  ```
- Se desejar reativar o TLS V1 CipherSpecs descontinuado, faça isso incluindo uma instrução DD simulada denominada TLS10ON (ative o TLS V1.0) na JCL do iniciador de canais. Se especificada por si só, isso ativa os CipherSpecs Fortes associados ao protocolo TLS 1.0:

  ```
  //TLS10ON DD DUMMY
  ```

  Se especificado com CSQXWEAK, isso também ativa o CipherSpecs **fraco** associado ao TLS 1.0.

- Se você deseja desativar explicitamente o CipherSpecs TLS V1 descontinuado, faça isso incluindo uma instrução DD simulada chamada TLS10OFF (desative o TLS V1.0) na JCL do iniciador de canais; por exemplo:

```
//TLS10OFF DD DUMMY
```

Se desejar negociar apenas com o listener usando as especificações de cifra listadas na lista de especificações de cifra padrão **System SSL** , será necessário definir a instrução DD a seguir na JCL CHINIT:

```
JCL: //GSKDCIPS DD DUMMY
```

Importante: Para o IBM MQ for z/OS 9.2.0 e mais recente, os cartões DD listados anteriormente e o valor de **AllowTLSV13** são levados em consideração ao exibir mensagens durante a inicialização do iniciador de canais para indicar quais protocolos estão ativados e quais não estão. Portanto, mesmo que um dos cartões DD listados anteriormente seja especificado, isso pode significar que, devido a uma combinação dessas configurações, um determinado protocolo não poderá ser ativado com outro protocolo. Por exemplo, o protocolo SSL 3.0 não será permitido se o TLS 1.3 estiver ativado.

Existem mecanismos alternativos que podem ser usados para reativar forçosamente os CipherSpecs fracos, e o suporte SSLv3, caso a mudança de Definição de Dados seja inapropriada. Entre em contato com o Serviço IBM para obter informações adicionais.

Conceitos relacionados

“Certificados digitais e compatibilidade de CipherSpec no IBM MQ” na página 48
Este tópico fornece informações sobre como escolher os certificados digitais e do CipherSpecs para sua política de segurança, delineando o relacionamento entre os certificados digitais e de CipherSpecs no IBM MQ.

Referências relacionadas

DEFINE CHANNEL

ALTER CHANNEL

Relacionamento entre as configurações do CipherSpec de alias

Estas informações descrevem o comportamento esperado com diferentes combinações de CipherSpecs de alias nas configurações do cliente e do servidor. Aqui, um cliente se refere à entidade que inicia a comunicação, por exemplo, um aplicativo cliente ou um canal emissor do gerenciador de filas; e o servidor refere-se à entidade que recebe a comunicação do cliente, por exemplo, um canal de conexão do servidor ou um canal receptor.

CipherSpecs de protocolo mínimo versus de protocolo fixo

O IBM MQ suporta dois tipos diferentes de CipherSpecs:

Protocolo mínimo
: Os CipherSpecs de protocolo mínimo são aqueles que não configuram um limite máximo, por exemplo, ANY, ANY_TLS12_OR_HIGHER ou ANY_TLS13_OR_HIGHER.

Protocolo fixo
: Os CipherSpecs de protocolo fixo são aqueles que identificam um protocolo específico, por exemplo, ANY_TLS12 e ANY_TLS13; ou um algoritmo específico como o ECDHE_ECDSA_3DES_EDE_CBC_SHA256.

Protocolo mínimo e fixo CipherSpecs são suportados em todas as plataformas.

Para maximizar a simplicidade de configuração e manter a segurança, o uso de CipherSpecs de **protocolo mínimo** é recomendado em ambos os lados do canal. Isso permite que suas comunicações suportem e usem automaticamente uma versão mais alta do protocolo TLS quando os dois lados suportam uma nova versão sem a necessidade de mudar a configuração dos dois lados.

Usando um CipherSpec de **protocolo mínimo** no lado de início, mas um CipherSpec de **protocolo fixo** no lado do receptor pode resultar na rejeição da conexão e

- Multi Na emissão das mensagens AMQ9631 e AMQ9641.
- z/OS Na emissão de mensagens CSQX631E e CSQX641E.

As tabelas a seguir mostram o relacionamento entre diferentes configurações de CipherSpec de alias e o resultado esperado. Tabela 81 na página 449 mostra o comportamento esperado quando o TLS 1.3 não está ativado no cliente, no servidor ou em ambos. A Tabela 82 na página 449 mostra o comportamento esperado quando o TLS 1.3 está ativado no cliente e no servidor Em ambos os casos, o CipherSpecs para o cliente é mostrado no eixo Y da tabela e o CipherSpecs para o servidor é mostrado no eixo X da tabela.

Nota: Nas tabelas a seguir, as células marcadas *Com probabilidade de falha* indicam o potencial para conflito quando você especifica um CipherSpec de **protocolo mínimo** para uma parte da conexão e um CipherSpec específico (**protocolo fixo**) para a outra parte.

Por exemplo, suponha que o cliente e o servidor estejam configurados para usar o CipherSpec ANY e o canal do servidor esteja configurado para usar um CipherSpec específico:

- Se o CipherSpec suportado mais forte tanto do cliente quanto do servidor corresponder ao CipherSpec específico configurado no canal, o handshake TLS será resolvido com sucesso.
- Se, no entanto, houver um CipherSpec mais forte que o suporte do cliente e do servidor e o handshake TLS resolver usá-lo, mesmo que ele não corresponda ao CipherSpec especificado no canal, o handshake TLS falhará.

Tabela 81. Comportamento esperado quando o TLS 1.3 não está ativado no cliente, no servidor ou em ambos

	Servidor			
Client	**CipherSpec específico do TLS 1.2**	**QUALQUER**	**ANY_ TLS12**	**ANY_TLS12_ OR_HIGHER**
CipherSpec específico do TLS 1.2	Connects	Connects	Connects	Connects
qualquer um	*Com probabilidade de falha*	Connects	Connects	Connects
ANY_ TLS12	*Com probabilidade de falha*	Connects	Connects	Connects
ANY_TLS12_ OR_HIGHER	*Com probabilidade de falha*	Connects	Connects	Connects

Tabela 82. Comportamento esperado quando o TLS 1.3 está ativado no cliente e no servidor

	Servidor						
Client	**CipherSpec específico do TLS 1.2**	**CipherSpec específico do TLS 1.3**	**QUALQUER**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ OR_HIGHER**	**ANY_TLS13_ OR_HIGHER**
CipherSpec específico do TLS 1.2	Connects	**Com falhas**	Connects	Connects	**Com falhas**	Connects	**Com falhas**
CipherSpec específico do TLS 1.3	**Com falhas**	Connects	Connects	**Com falhas**	Connects	Connects	Connects

Tabela 82. Comportamento esperado quando o TLS 1.3 está ativado no cliente e no servidor (continuação)							
	Servidor						
Client	**CipherSpec específico do TLS 1.2**	**CipherSpec específico do TLS 1.3**	**QUALQUER**	**ANY_TLS12**	**ANY_TLS13**	**ANY_TLS12_OR_HIGHER**	**ANY_TLS13_OR_HIGHER**
qualquer um	**Com falhas**	*Com probabilidade de falha*	Connects	**Com falhas**	Connects	Connects	Connects
ANY_TLS12	*Com probabilidade de falha*	**Com falhas**	Connects	Connects	**Com falhas**	Connects	**Com falhas**
ANY_TLS13	**Com falhas**	*Com probabilidade de falha*	Connects	**Com falhas**	Connects	Connects	Connects
ANY_TLS12_OR_HIGHER	**Com falhas**	*Com probabilidade de falha*	Connects	**Com falhas**	Connects	Connects	Connects
ANY_TLS13_OR_HIGHER	**Com falhas**	*Com probabilidade de falha*	Connects	**Com falhas**	Connects	Connects	Connects

Conceitos relacionados

"Certificados digitais e compatibilidade de CipherSpec no IBM MQ" na página 48
Este tópico fornece informações sobre como escolher os certificados digitais e do CipherSpecs para sua política de segurança, delineando o relacionamento entre os certificados digitais e de CipherSpecs no IBM MQ.

"CipherSpecs e CipherSuites" na página 22
Os protocolos de segurança criptográficos devem concordar sobre os algoritmos usados por uma conexão segura. CipherSpecs e CipherSuites definem combinações específicas de algoritmos.

"Ativando CipherSpecs" na página 427
Ative um CipherSpec usando o parâmetro **SSLCIPH** no comando do MQSC **DEFINE CHANNEL** ou **ALTER CHANNEL**.

Tarefas relacionadas

Migrando configurações de segurança existentes para usar o CipherSpec ANY_TLS12_OR_HIGHER

Obtendo informações sobre CipherSpecs usando o IBM MQ Explorer

É possível usar o IBM MQ Explorer para exibir descrições de CipherSpecs.

Utilize o seguinte procedimento para obter informações sobre o CipherSpecs no "Ativando CipherSpecs" na página 427:

1. Abra o IBM MQ Explorer e expanda a pasta **Gerenciadores de Filas**.
2. Certifique-se de que iniciou o gerenciador de filas.
3. Selecione o Gerenciador de Filas com o qual deseja trabalhar e clique em **Canais**.
4. Clique com o botão direito no canal que deseja trabalhar e selecione **Propriedades**.
5. Selecione a página de propriedades **SSL**.
6. Selecione da lista o CipherSpec com o qual quer trabalhar. Uma descrição é exibida na janela abaixo da lista.

z/OS IBM i Alternativas para a Especificação do CipherSpecs

Para aquelas plataformas em que o sistema operacional fornece suporte do TLS, é possível que o seu sistema suporte os novos CipherSpecs que não estão incluídos em “Ativando CipherSpecs” na página 427.

Você pode especificar um novo CipherSpec com o parâmetro SSLCIPH, mas o valor que você fornecer dependerá da sua plataforma. Em todos os casos, a especificação deve corresponder a um CipherSpec do TLS que seja válido e suportado pela versão de TLS que seu sistema está executando.

Nota: Esta seção não se aplica a sistemas AIX, Linux, and Windows, pois os CipherSpecs são fornecidos com o produto IBM MQ, portanto, os novos CipherSpecs não são disponibilizados após o envio.

IBM i IBM i

Uma cadeia de dois caracteres que representa um valor hexadecimal.

Para obter mais informações sobre os valores permitidos, consulte o ponto três na seção Notas de uso de Configurar informações de caracteres para uma sessão segura.

Atenção: Não é necessário especificar valores de cifras hexadecimais no **SSLCIPH**, já que o valor não especifica claramente qual cifra será usada e a opção de qual protocolo será usado está indeterminada. A utilização de valores de cifras hexadecimais pode levar a erros de incompatibilidade de CipherSpec.

É possível usar o comando **CHGMQMCHL** ou **CRTMQMCHL** para especificar o valor, por exemplo:

```
CRTMQMCHL CHLNAME(' channel name ') SSLCIPH(' hexadecimal value ')
```

Também é possível usar o comando do MQSC **ALTER QMGR** para configurar o parâmetro **SSLCIPH**.

z/OS z/OS

Uma sequência de quatro caracteres que representa um valor hexadecimal. Os códigos hexadecimais correspondem aos valores definidos no protocolo TLS.

Para obter mais informações, consulte as Definições do conjunto de cifras onde há uma lista de todas as especificações de cifras TLS 1.0, TLS 1.2 e TLS 1.3 na forma de códigos hexadecimais de 4 dígitos.

Nota: Deprecated Para usar um CipherSpec fraco, ou um CipherSpec pertencente a um protocolo descontinuado, como o SSL V3.0 ou o TLS 1.0, deve-se especificar o cartão DD relevante no JCL de inicialização do inicializador de canais. Consulte “CipherSpecs descontinuado” na página 443 para obter mais informações.

Considerações para clusters do IBM MQ

Com clusters do IBM MQ é mais seguro usar os nomes de CipherSpec no “Ativando CipherSpecs” na página 427. Se você usar uma especificação alternativa, tenha em mente que a especificação pode não ser válida em outras plataformas. Para obter informações adicionais, consulte “SSL/TLS e clusters” na página 490.

Especificando um CipherSpec para um IBM MQ MQI client

Você tem três opções para especificar um CipherSpec para um IBM MQ MQI client.

Estas opções são as seguintes:

- Utilizando uma tabela de definição de canais
- Usando o campo SSLCipherSpec na estrutura MQCD, no MQCD_VERSION_7 ou mais recente, em uma chamada MQCONNX.
- Usando o Active Directory (em sistemas Windows com suporte ao Active Directory)

Especificando um CipherSuite com o IBM MQ classes for Java e IBM MQ classes for JMS

O IBM MQ classes for Java e o IBM MQ classes for JMS especificam CipherSuites diferentemente de outras plataformas.

Para obter informações sobre como especificar um CipherSuite com o IBM MQ classes for Java, consulte Suporte de Segurança da Camada de Transporte (TLS) para o Java

Para obter informações sobre como especificar um CipherSuite com o IBM MQ classes for JMS, consulte Usando a Segurança da Camada de Transporte (TLS) com o IBM MQ classes for JMS

Especificando um CipherSpec para IBM MQ.NET

Para o IBM MQ.NET é possível especificar o CipherSpec usando a classe MQEnvironment ou usando o MQC.SSL_CIPHER_SPEC_PROPERTY na hashtable das propriedades da conexão.

Para obter informações sobre como especificar um CipherSpec para o cliente não gerenciado do .NET, consulte Ativando o TLS para o cliente não gerenciado do .NET

Para obter informações sobre como especificar um CipherSpec para o cliente gerenciado do .NET, consulte Suporte de CipherSpec para cliente gerenciado do .NET

z/OS Uso de AT-TLS com o IBM MQ for z/OS

A Segurança da Camada de Transporte Transparente do Aplicativo (AT-TLS) fornece suporte de TLS para aplicativos z/OS sem que esses aplicativos tenham que implementar o suporte de TLS, ou até mesmo saber que o TLS está sendo usado. O AT-TLS está disponível somente no z/OS.

O AT-TLS pode ser usado com todas as versões do IBM MQ for z/OS.

Antes de fazer uso de AT-TLS com o IBM MQ for z/OS, certifique-se de entender o “Restrições” na página 455 envolvido.

Para usar a Segurança da Camada de Transporte Transparente do Aplicativo, defina instruções de política contendo um conjunto de regras que são usadas pelo z/OS Communications Server para decidir quais conexões TCP/IP têm o TLS ativado de forma transparente.

O IBM MQ for z/OS tem sua própria implementação TLS, que requer que os canais tenham o parâmetro SSLCIPH configurado com uma CipherSpec suportada.

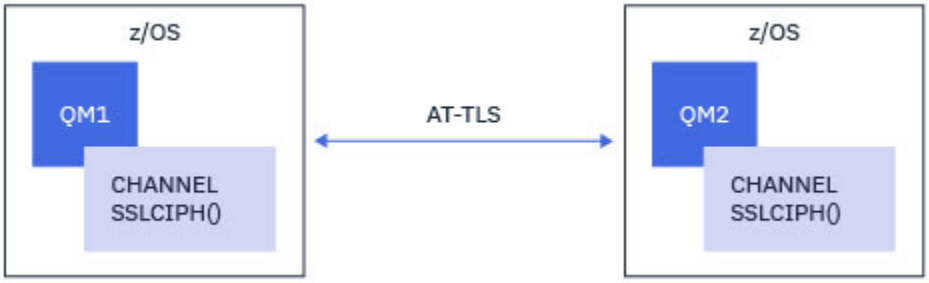
Ao decidir ativar o TLS em um canal, o administrador do IBM MQ pode decidir usar o AT-TLS ou o TLS do IBM MQ. A decisão é frequentemente feita com base em se o AT-TLS é usado para outro middleware ou por causa de implicações de desempenho. Para uma comparação básica do desempenho do AT-TLS e do TLS do IBM MQ, consulte MP16: Planejamento de capacidade e ajuste para o IBM MQ for z/OS.

Cenários

O uso do AT-TLS com o IBM MQ é suportado nos cenários a seguir:

Cenário 1

Entre dois Gerenciadores de Filas do IBM MQ for z/OS em que ambos os lados do canal usam AT-TLS. Ou seja, nenhum canal especifica o atributo SSLCIPH. Esta abordagem pode ser usada com qualquer canal de mensagem.



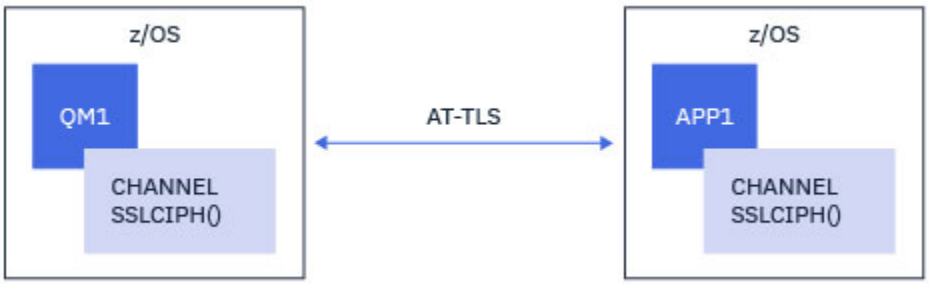
A implementação deste cenário consiste na definição de duas políticas de AT-TLS, uma para cada lado do canal. Essas políticas são as mesmas usadas com o Cenário 3 ou Cenário 4.

Por exemplo, se o canal estivesse mudando do uso de uma única CipherSpec denominada para o AT-TLS, o canal de saída usaria a política do “Configurando o AT-TLS em um canal de saída por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec denominado” na página 456 e o canal de entrada usaria a política do “Configurando o AT-TLS em um canal de entrada por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec de nominado” na página 465.

Se o canal estivesse mudando do uso de uma CipherSpec de alias para o AT-TLS, o canal de saída usaria a política do “Configurando o AT-TLS em um canal de saída para um Gerenciador de Filas IBM MQ for Multiplatforms usando um alias CipherSpecs” na página 461 e o canal de entrada usaria a política do “Configurando o AT-TLS em um canal de entrada para um Gerenciador de Filas IBM MQ for Multiplatforms usando um CipherSpecs de alias” na página 469.

Cenário 2

Entre um Gerenciador de Filas do IBM MQ for z/OS e um aplicativo cliente IBM MQ Java executando no z/OS em que ambos os lados do canal usam AT-TLS. Ou seja, nem o canal de conexão do servidor, nem o canal de conexão do cliente especifica o atributo SSLCIPH.



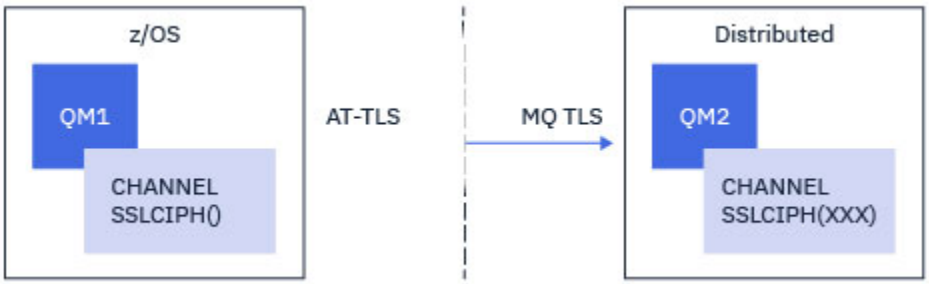
A implementação deste cenário consiste na definição de duas políticas de AT-TLS, uma para cada lado do canal. Essas políticas são as mesmas usadas com o Cenário 3 ou Cenário 4.

Por exemplo, se o canal estivesse mudando do uso de uma única CipherSpec denominada para o AT-TLS, o canal de conexão do cliente usaria a política do “Configurando o AT-TLS em um canal de saída por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec denominado” na página 456 e o canal de conexão do servidor usaria a política do “Configurando o AT-TLS em um canal de entrada por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec de nominado” na página 465.

Se o canal estivesse mudando do uso de uma CipherSpec de alias para o AT-TLS, o canal de conexão do cliente usaria a política do “Configurando o AT-TLS em um canal de saída para um Gerenciador de Filas IBM MQ for Multiplatforms usando um alias CipherSpecs” na página 461 e o canal de conexão do servidor usaria a política do “Configurando o AT-TLS em um canal de entrada para um Gerenciador de Filas IBM MQ for Multiplatforms usando um CipherSpecs de alias” na página 469.

Cenário 3

Entre umIBM MQ for z/OS gerenciador de filas e um gerenciador de filas em execuçãoIBM MQ for Multiplatforms , onde oIBM MQ for z/OS gerenciador de filas usa AT-TLS e oIBM MQ for Multiplatforms gerenciador de filas usaIBM MQ TLS, especificando o atributo SSLCIPH com um único nomeCipherSpec . Isto se aplica a todos os tipos de canal de mensagens diferentes do emissor de cluster e do receptor de cluster.

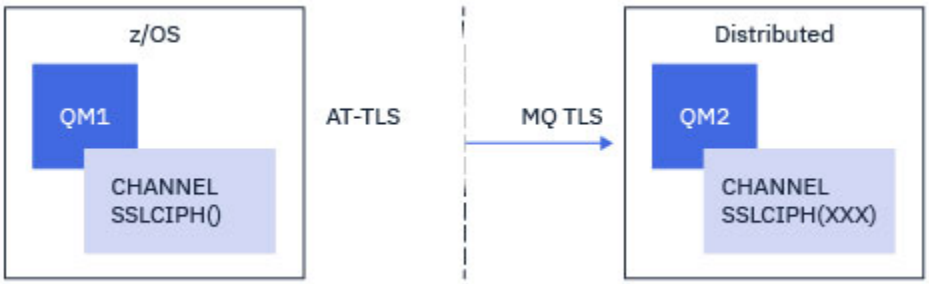


Consulte “Configurando o AT-TLS em um canal de saída por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec denominado” na página 456 para um exemplo de configuração de AT-TLS para canais de saída do Gerenciador de Filas do IBM MQ for z/OS para o Gerenciador de Filas do IBM MQ for Multiplatforms e “Configurando o AT-TLS em um canal de entrada por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec de nominado” na página 465 para um exemplo de configuração de AT-TLS para canais de entrada do Gerenciador de Filas do IBM MQ for Multiplatforms para o Gerenciador de Filas do IBM MQ for z/OS.

A mesma configuração de AT-TLS pode ser usada quando ambos os Gerenciadores de Filas estão no z/OS, mas o Gerenciador de Filas no lado direito não foi configurado para usar o AT-TLS.

Cenário 4

Entre um Gerenciador de Filas do IBM MQ for z/OS e um Gerenciador de Filas em execução no IBM MQ for Multiplatforms, em que o Gerenciador de Filas do IBM MQ for z/OS usa o AT-TLS e o Gerenciador de Filas do IBM MQ for Multiplatforms usa o TLS do IBM MQ, especificando o atributo SSLCIPH com um CipherSpec de alias. Isto se aplica a todos os tipos de canal de mensagens diferentes do emissor de cluster e do receptor de cluster.

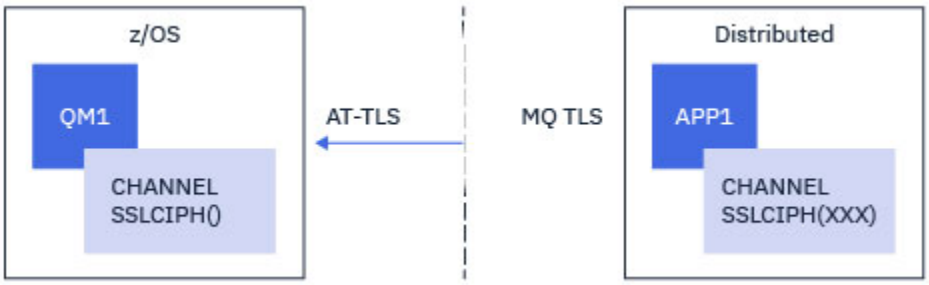


Consulte “Configurando o AT-TLS em um canal de saída para um Gerenciador de Filas IBM MQ for Multiplatforms usando um alias CipherSpecs” na página 461 para um exemplo de configuração de AT-TLS para canais de saída do Gerenciador de Filas do IBM MQ for z/OS para o Gerenciador de Filas do IBM MQ for Multiplatforms, e “Configurando o AT-TLS em um canal de entrada para um Gerenciador de Filas IBM MQ for Multiplatforms usando um CipherSpecs de alias” na página 469 e “Configurando o AT-TLS em um canal de entrada para um Gerenciador de Filas IBM MQ for Multiplatforms usando um CipherSpecs de alias” na página 469 para um exemplo de configuração de AT-TLS para canais de entrada do Gerenciador de Filas do IBM MQ for Multiplatforms para o Gerenciador de Filas do IBM MQ for z/OS.

A mesma configuração de AT-TLS pode ser usada quando ambos os Gerenciadores de Filas estão no z/OS, mas o Gerenciador de Filas no lado direito não foi configurado para usar o AT-TLS.

Cenário 5

Entre um Gerenciador de Filas do IBM MQ for z/OS e um aplicativo cliente em execução no IBM MQ for Multiplatforms, em que o Gerenciador de Filas do IBM MQ for z/OS usa AT-TLS e o aplicativo cliente usa o TLS do IBM MQ especificando o atributo SSLCIPH com uma CipherSpec única denominada.

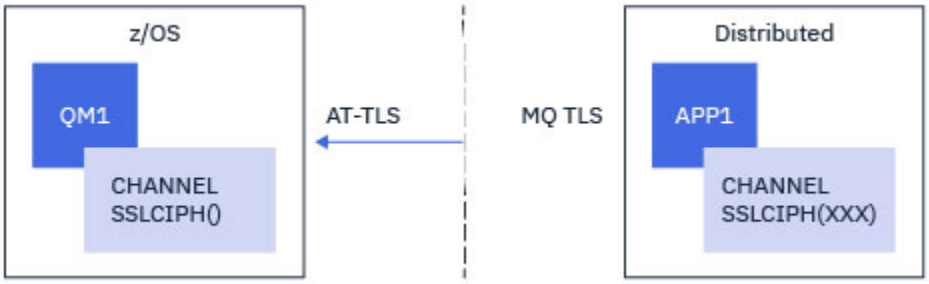


Este cenário requer uma única política AT-TLS que atenda aos mesmos requisitos dos utilizados por um canal de mensagem de entrada; consulte "Configurando o AT-TLS em um canal de entrada por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec de nominado" na página 465.

A mesma configuração de AT-TLS pode ser usada quando o aplicativo cliente é um aplicativo Java e também está em execução no z/OS, mas não foi configurado para usar o AT-TLS.

Cenário 6

Entre um Gerenciador de Filas do IBM MQ for z/OS e um aplicativo cliente em execução no IBM MQ for Multiplatforms, em que o Gerenciador de Filas do IBM MQ for z/OS usa AT-TLS e o aplicativo cliente usa o TLS do IBM MQ especificando o atributo SSLCIPH com uma CipherSpec de alias.



Este cenário requer uma única política AT-TLS que atenda aos mesmos requisitos dos utilizados por um canal de mensagem de entrada; consulte "Configurando o AT-TLS em um canal de entrada para um Gerenciador de Filas IBM MQ for Multiplatforms usando um CipherSpecs de alias" na página 469.

A mesma configuração de AT-TLS pode ser usada quando o aplicativo cliente é um aplicativo Java e também está em execução no z/OS, mas não foi configurado para usar o AT-TLS.

Restrições

O IBM MQ for z/OS não é ciente do AT-TLS, portanto, existem várias restrições que se aplicam com os cenários precedentes:

- O AT-TLS em combinação com o TLS do IBM MQ não funciona com canais emissores e receptores de cluster.
- Os Gerenciadores de Filas do IBM MQ for z/OS não são cientes de que estão usando o AT-TLS e não recebem nenhuma informação de certificado de seu Gerenciador de Filas ou cliente parceiro. Portanto, os atributos a seguir não têm efeito no lado do z/OS de um canal usando o AT-TLS:

  - Os atributos de canal SSLCAUTH e SSLPEER
  - O atributo SSLRKEYC do Gerenciador de Filas
  - Os atributos SSLPEERMAP de regras CHLAUTH
- O uso da renegociação de chave secreta do TLS requer que ambos os lados do canal usem o TLS do IBM MQ. Portanto, um Gerenciador de Filas do IBM MQ for Multiplatforms, ou cliente, não deve ter a renegociação de chave secreta do TLS ativada se estiver se conectando a um Gerenciador de Filas do IBM MQ for z/OS usando o AT-TLS.

  Para desativar a renegociação de chave secreta TLS para um gerenciador de filas, configure o parâmetro SSLRKEYC do gerenciador de filas para `0`. Para um cliente, configure o parâmetro relevante para `0` dependendo do tipo do cliente. Para obter detalhes sobre como fazer isso, consulte "Reconfigurando as chaves secretas SSL e TLS" na página 474.

Instruções de configuração de AT-TLS

O AT-TLS é configurado usando um conjunto de instruções. Os utilizados nos cenários documentados neste tópico são:

TTLSRule
Especifica um conjunto de critérios para correspondência de uma conexão TCP/IP com uma configuração de TLS. Isto, por sua vez, refere-se aos outros tipos de instrução.

TTLSGroupAction
Especifica se a `TTLSRule` de referência está ativado ou não.

TTLSEnvironmentAction
Especifica a configuração detalhada para a `TTLSRule` de referência e referencia uma série de outras instruções.

TTLSKeyringParms
Referencia o conjunto de chaves que deve ser usado pelo AT-TLS.

TTLSCipherParms
Define os conjuntos de cifras que devem ser usados.

TTLSEnvironmentAdvancedParms
Define quais protocolos TLS ou SSL estão ativados.

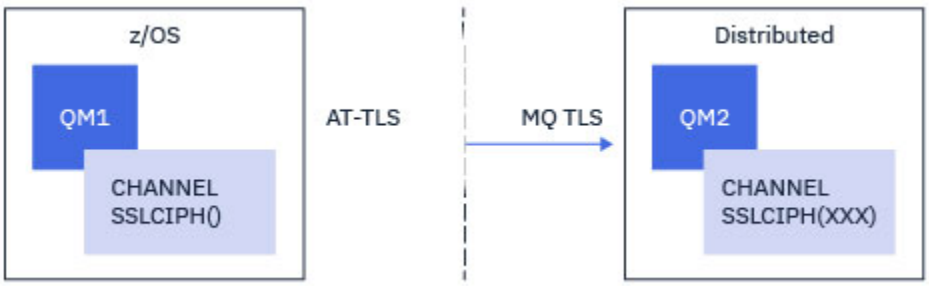
Atenção: Há outras Instruções de política AT-TLS com AT-TLS que não são documentadas aqui, e poderiam ser usadas com o IBM MQ dependendo da necessidade. No entanto, o IBM MQ somente foi testado com as políticas descritas neste tópico.

z/OS *Configurando o AT-TLS em um canal de saída por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec denominado*

Como você configura o AT-TLS em um canal de saída por meio de um Gerenciador de Filas IBM MQ for z/OS para um Gerenciador de Filas IBM MQ for Multiplatforms. Neste caso, o canal no Gerenciador de Filas z/OS é um canal emissor que não possui o conjunto de atributos SSLCIPH, e o canal no Gerenciador de Filas não z/OS é um canal receptor com o conjunto de atributos SSLCIPH configurado para um único CipherSpec denominado.

Consulte "Configurando o AT-TLS em um canal de saída para um Gerenciador de Filas IBM MQ for Multiplatforms usando um alias CipherSpecs" na página 461 para um exemplo usando um CipherSpec de alias.

Neste exemplo, um par de canais emissor-receptor existente, que usa a CipherSpec TLS 1.3 TLS_AES_256_GCM_SHA384 será ajustado para que o canal emissor use o AT-TLS em vez de IBM MQ TLS.



Outros protocolos TLS e CipherSpecs podem ser usados fazendo pequenos ajustes na configuração. Outros tipos de canal de mensagens, além de canais emissores e receptores de cluster, poderiam ser usados sem alteração na configuração de AT-TLS.

Procedimento

Etapa 1: Pare o canal

Etapa 2: Crie e aplique uma política AT-TLS

É necessário criar as instruções AT-TLS a seguir para este cenário:

1. Uma instrução TTLSRule para corresponder as conexões de saída do espaço de endereço do iniciador do canal para o endereço IP e número da porta do canal receptor de destino. Esses valores devem corresponder às informações utilizadas no CONNAME do canal emissor. Aqui, mais filtragem foi incluída para corresponder a um nome de tarefa do iniciador de canal específico.

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

   A regra anterior corresponde às conexões que vão para o endereço IP 123.456.78.9 na porta 1414 por meio da tarefa CSQ1CHIN.

   Mais opções avançadas de filtragem são descritas em TTLSRule.

2. Uma instrução TTLSGroupAction ativando a regra. O `TTLSRule` referencia o `TTLSGroupAction` usando a propriedade **`TTLSGroupActionRef`**.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Uma instrução TTLSEnvironmentAction está associada com o `TTLSRule` pela propriedade **`TTLSEnvironmentActionRef`**. Um `TTLSEnvironmentAction` configura o Ambiente TLS e especifica qual conjunto de chaves a utilizar.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. Uma instrução TTLSKeyringParms é associada ao TTLSEnvironmentAction pela propriedade **TTLSKeyringParmsRef** e define o conjunto de chaves usado por AT-TLS.

   O conjunto de chaves deve conter certificados confiáveis pelo Gerenciador de Filas remoto não z/OS. Este conjunto de chaves pode ser definido da mesma forma que um conjunto de chaves usado pelo inicializador de canal; consulte "Configuring your z/OS system to use TLS" na página 259.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Uma instrução TTLSCipherParms associada com o TTLSEnvironmentAction pela propriedade **TTLSCipherParmsRef**

   Esta instrução deve conter um único nome de conjunto de cifras que deve ser o equivalente ao nome de CipherSpec do IBM MQ usado no canal receptor de destino.

   Nota: Os nomes de conjunto de cifras AT-TLS não correspondem necessariamente aos nomes de CipherSpec do IBM MQ. No entanto, é possível localizar o nome do conjunto de cifras AT-TLS que corresponde a um nome de CipherSpec do IBM MQ localizando o nome de CipherSpec do IBM MQ na tabela a seguir e fazendo referência cruzada da coluna de código hexadecimal com a coluna de caracteres expandida da Tabela 2 no tópico de instrução TTLSCipherParms.

Tabela 83. CipherSpecs no z/OS do IBM MQ for z/OS 9.2.0

CipherSpec	Protocolo	Código Hexadecimal	Ativado por padrão
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sim
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sim
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sim
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sim
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sim
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sim
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sim
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sim
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sim
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sim

Tabela 83. CipherSpecs no z/OS do IBM MQ for z/OS 9.2.0 (continuação)			
CipherSpec	**Protocolo**	**Código Hexadecimal**	**Ativado por padrão**
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sim
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sim
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sim
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	No
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	No
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	No
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	No
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	No
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	No
TRIPLE_DES_SHA_US	SSL v3	000A	No
RC4_SHA_US	SSL v3	0005	No
RC4_MD5_US	SSL v3	0004	No
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	No
RC2_MD5_EXPORT	SSL v3	0006	No
NULL_SHA	SSL v3	0002	No
NULL_MD5	SSL v3	0001	No

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. Uma instrução TTLSEnvironmentAdvancedParms é associada ao `TTLSEnvironmentAction` pela propriedade **`TTLSEnvironmentAdvancedParmsRef`** ..

   Essa instrução pode ser usada para especificar quais protocolos SSL e TLS estão ativados. Com IBM MQ você deve ativar apenas o protocolo único que corresponde ao nome do conjunto de cifras usado na instrução `TTLSCipherParms` .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

O conjunto completo de instruções é o seguinte e deve ser aplicado ao agente de política:

```
TTLSRule                      CSQ1-TO-REMOTE
{
  LocalAddr                   ALL
  RemoteAddr                  123.456.78.9
  RemotePortRange             1414
  Jobname                     CSQ1CHIN
  Direction                   OUTBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   CLIENT
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Etapa 3: Remover SSLCIPH do z/OS canal

Remova o CipherSpec do canal z/OS usando o comando a seguir:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Etapa 4: Iniciar o canal

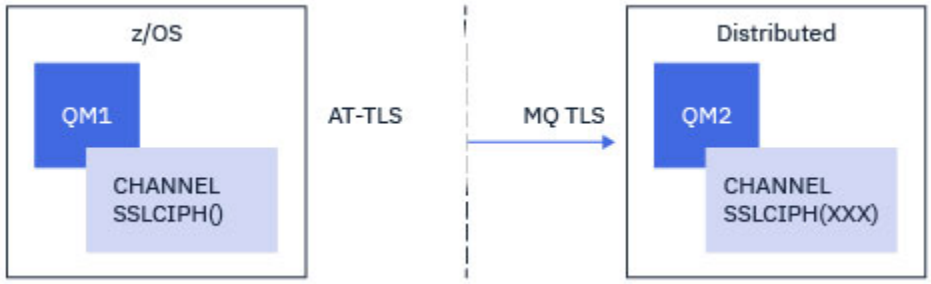
Uma vez que o canal tenha iniciado, ele estará usando uma combinação de AT-TLS e TLS do IBM MQ.

Atenção: As instruções AT-SLT anteriores são apenas uma configuração mínima Há outras Instruções de política AT-TLS com AT-TLS que não são documentadas aqui, e poderiam ser usadas com o IBM MQ dependendo da necessidade. No entanto, o IBM MQ foi testado apenas com as políticas descritas

z/OS *Configurando o AT-TLS em um canal de saída para um Gerenciador de Filas IBM MQ for Multiplatforms usando um alias CipherSpecs*

Como você configura o AT-TLS em um canal de saída por meio de um Gerenciador de Filas IBM MQ for z/OS para um Gerenciador de Filas IBM MQ for Multiplatforms. Neste caso, o canal no Gerenciador de Filas z/OS é um canal emissor que não possui o conjunto de atributos SSLCIPH, e o canal no Gerenciador de Filas não z/OS é um canal receptor com o atributo SSLCIPH configurado para um alias CipherSpec

Neste exemplo, um par de canais emissor-receptor existente, que usa o CipherSpec de alias ANY_TLS13, será ajustado para que o canal emissor utilize AT-TLS em vez de IBM MQ TLS.



Outros protocolos TLS e CipherSpecs podem ser usados fazendo pequenos ajustes na configuração. Outros tipos de canal de mensagens, além de canais emissores e receptores de cluster, poderiam ser usados sem alteração na configuração de AT-TLS.

Procedimento

Etapa 1: Pare o canal

Etapa 2: Crie e aplique uma política AT-TLS

É necessário criar as instruções AT-TLS a seguir para este cenário:

1. Uma instrução TTLSRule para corresponder as conexões de saída do espaço de endereço do iniciador do canal para o endereço IP e número da porta do canal receptor de destino. Esses valores devem corresponder às informações utilizadas no CONNAME do canal emissor. Aqui, mais filtragem foi incluída para corresponder a um nome de tarefa do iniciador de canal específico.

```
TTLSRule                        CSQ1-TO-REMOTE
{
  LocalAddr                     ALL
  RemoteAddr                    123.456.78.9
  RemotePortRange               1414
  Jobname                       CSQ1CHIN
  Direction                     OUTBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}
```

   A regra anterior corresponde às conexões que vão para o endereço IP 123.456.78.9 na porta 1414 por meio da tarefa CSQ1CHIN.

   Mais opções avançadas de filtragem são descritas em TTLSRule.

2. Uma instrução TTLSGroupAction ativando a regra. O `TTLSRule` referencia o `TTLSGroupAction` usando a propriedade **`TTLSGroupActionRef`**.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Uma instrução TTLSEnvironmentAction está associada com o TTLSRule pela propriedade **TTLSEnvironmentActionRef**. Um TTLSEnvironmentAction configura o Ambiente TLS e especifica qual conjunto de chaves a utilizar.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. Uma instrução TTLSKeyringParms é associada ao TTLSEnvironmentAction pela propriedade **TTLSKeyringParmsRef** e define o conjunto de chaves usado por AT-TLS.

   O conjunto de chaves deve conter certificados confiáveis pelo Gerenciador de Filas remoto não z/OS. Este conjunto de chaves pode ser definido da mesma forma que um conjunto de chaves usado pelo inicializador de canal; consulte “Configuring your z/OS system to use TLS” na página 259.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Uma instrução TTLSCipherParms associada com o TTLSEnvironmentAction pela propriedade **TTLSCipherParmsRef**

   Esta instrução deve conter um ou mais nomes de conjunto de cifras, pelo menos um dos quais deve ser compatível com o conjunto de CipherSpecs implicado pelo CipherSpec de alias usado no canal do receptor de destino.

   Nota: Os nomes de conjunto de cifras AT-TLS não correspondem necessariamente aos nomes de CipherSpec do IBM MQ. No entanto, é possível localizar o nome do conjunto de cifras AT-TLS que corresponde a um nome de CipherSpec do IBM MQ localizando o nome de CipherSpec do IBM MQ na tabela a seguir e fazendo referência cruzada da coluna de código hexadecimal com a coluna de caracteres expandida da Tabela 2 no tópico TTLSCipherParms.

Tabela 84. CipherSpecs no z/OS do IBM MQ for z/OS 9.2.0

CipherSpec	Protocolo	Código Hexadecimal	Ativado por padrão
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sim
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sim
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sim
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sim
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sim
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sim
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sim
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sim

Tabela 84. CipherSpecs no z/OS do IBM MQ for z/OS 9.2.0 (continuação)			
CipherSpec	**Protocolo**	**Código Hexadecimal**	**Ativado por padrão**
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sim
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sim
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sim
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sim
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sim
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	No
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	No
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	No
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	No
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	No
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	No
TRIPLE_DES_SHA_US	SSL v3	000A	No
RC4_SHA_US	SSL v3	0005	No
RC4_MD5_US	SSL v3	0004	No
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	No
RC2_MD5_EXPORT	SSL v3	0006	No
NULL_SHA	SSL v3	0002	No
NULL_MD5	SSL v3	0001	No

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

Atenção: Se o gerenciador de filas e a política de AT-TLS suportarem TLS 1.3, somente os CipherSpecs de alias que contêm pelo menos um TLS 1.3 CipherSpec permitirão que o canal seja iniciado. Por exemplo, usar ANY_TLS12 resulta no canal falhando ao iniciar, mesmo se TTLSCipherParms contiver TLS 1.2 CipherSpecs, mas usando ANY_TLS12_OR_HIGHER ou ANY_TLS13 permite que o canal seja iniciado. Consulte “Relacionamento entre as configurações do CipherSpec de alias” na página 448 para uma explicação.

6. Uma instrução TTLSEnvironmentAdvancedParms é associada ao TTLSEnvironmentAction pela propriedade **TTLSEnvironmentAdvancedParmsRef** ..

   Essa instrução pode ser usada para especificar quais protocolos SSL e TLS estão ativados e deve ser consistente com os conjuntos de cifras na instrução TTLSCipherParms

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

O conjunto completo de instruções é o seguinte e deve ser aplicado ao agente de política:

```
TTLSRule                      CSQ1-TO-REMOTE
{
  LocalAddr                   ALL
  RemoteAddr                  123.456.78.9
  RemotePortRange             1414
  Jobname                     CSQ1CHIN
  Direction                   OUTBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-OUTBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   CLIENT
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Etapa 3: Remover SSLCIPH do z/OS canal

Remova o CipherSpec do canal z/OS usando o comando a seguir:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Etapa 4: Iniciar o canal

Uma vez que o canal tenha iniciado, ele estará usando uma combinação de AT-TLS e TLS do IBM MQ.

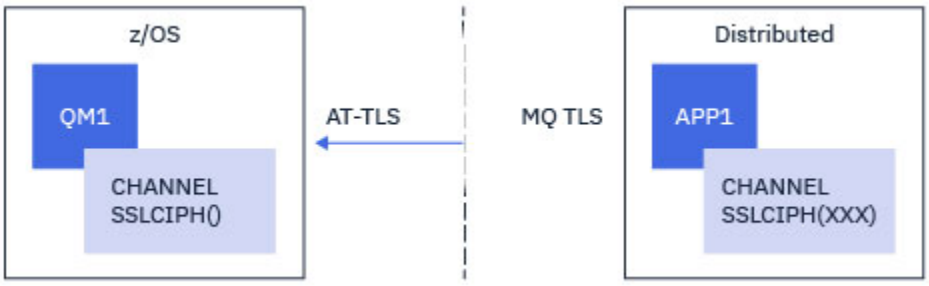
Atenção: As instruções AT-SLT anteriores são apenas uma configuração mínima Há outras Instruções de política AT-TLS com AT-TLS que não são documentadas aqui, e poderiam ser usadas com o IBM MQ dependendo da necessidade. No entanto, o IBM MQ foi testado apenas com as políticas descritas

z/OS *Configurando o AT-TLS em um canal de entrada por meio de um Gerenciador de Filas IBM MQ for Multiplatforms usando um único CipherSpec de nominado*

Como você configura o AT-TLS em um canal de entrada por meio de um Gerenciador de Filas IBM MQ for Multiplatforms para um Gerenciador de Filas IBM MQ for z/OS. Neste caso, o canal no Gerenciador de Filas z/OS é um canal receptor que não possui o conjunto de atributos SSLCIPH, e o canal no Gerenciador de Filas não z/OS é um canal emissor com o conjunto de atributos SSLCIPH configurado para um único CipherSpec denominado.

Consulte “Configurando o AT-TLS em um canal de entrada para um Gerenciador de Filas IBM MQ for Multiplatforms usando um CipherSpecs de alias” na página 469 para um exemplo usando um CipherSpec de alias.

Neste exemplo, um par de canais emissor-receptor existente, que usa a CipherSpec TLS 1.3 TLS_AES_256_GCM_SHA384 será ajustado para que o canal receptor use o AT-TLS em vez de IBM MQ TLS.



Outros protocolos TLS e CipherSpecs podem ser usados fazendo pequenos ajustes na configuração. Outros tipos de canal de mensagens, além de canais emissores e receptores de cluster, poderiam ser usados sem alteração na configuração de AT-TLS.

Procedimento

Etapa 1: Pare o canal

Etapa 2: Crie e aplique uma política AT-TLS

É necessário criar as instruções AT-TLS a seguir para este cenário:

1. Uma instrução TTLSRule para corresponder conexões de entrada com o espaço de endereço do iniciador do canal por meio do endereço IP do canal emissor. Aqui, mais filtragem foi incluída para corresponder a um nome de tarefa do iniciador de canal específico.

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

As correspondências de regra anteriores com relação às conexões que vão para a tarefa CSQ1CHIN na porta local 1414 do endereço IP remoto 123.456.78.9.

Mais opções avançadas de filtragem são descritas em TTLSRule.

2. Uma instrução TTLSGroupAction ativando a regra. O `TTLSRule` referencia o `TTLSGroupAction` usando a propriedade **`TTLSGroupActionRef`**.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Uma instrução TTLSEnvironmentAction está associada com o `TTLSRule` pela propriedade **`TTLSEnvironmentActionRef`**. Um `TTLSEnvironmentAction` configura o Ambiente TLS e especifica qual conjunto de chaves a utilizar.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

O AT-TLS fornece a capacidade de conceder autenticação mútua, que é o equivalente a utilizar o atributo de canal SSLCAUTH. Isso é feito por ter uma instrução `TTLSEnvironmentAction` com um valor **`HandshakeRole`** de *ServerWithClientAuth* para a instrução de entrada `TTLSEnvironmentAction`.

4. Uma instrução TTLSKeyringParms é associada ao `TTLSEnvironmentAction` pela propriedade **`TTLSKeyringParmsRef`** e define o conjunto de chaves usado por AT-TLS.

   O conjunto de chaves deve conter certificados confiáveis pelo Gerenciador de Filas remoto não z/OS. Este conjunto de chaves pode ser definido da mesma forma que um conjunto de chaves usado pelo inicializador de canal; consulte “Configuring your z/OS system to use TLS” na página 259.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Uma instrução TTLSCipherParms associada com o `TTLSEnvironmentAction` pela propriedade **`TTLSCipherParmsRef`**

   Esta instrução deve conter um único nome de conjunto de cifras que deve ser o equivalente ao nome de CipherSpec do IBM MQ usado no canal emissor remoto.

   Nota: Os nomes de conjunto de cifras AT-TLS não correspondem necessariamente aos nomes de CipherSpec do IBM MQ. No entanto, é possível localizar o nome do conjunto de cifras AT-TLS que corresponde a um nome de CipherSpec do IBM MQ localizando o nome de CipherSpec do IBM MQ na tabela a seguir e fazendo referência cruzada da coluna de código hexadecimal com a coluna de caracteres expandida da Tabela 2 no tópico de instrução TTLSCipherParms.

Tabela 85. CipherSpecs no z/OS do IBM MQ for z/OS 9.2.0			
CipherSpec	**Protocolo**	**Código Hexadecimal**	**Ativado por padrão**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sim
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sim
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sim
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sim
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sim
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sim
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sim
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sim
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sim
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sim
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sim
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sim
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sim
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	No
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	No
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	No
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	No
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	No
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	No
TRIPLE_DES_SHA_US	SSL v3	000A	No
RC4_SHA_US	SSL v3	0005	No
RC4_MD5_US	SSL v3	0004	No

Tabela 85. CipherSpecs no z/OS do IBM MQ for z/OS 9.2.0 (continuação)			
CipherSpec	**Protocolo**	**Código Hexadecimal**	**Ativado por padrão**
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	No
RC2_MD5_EXPORT	SSL v3	0006	No
NULL_SHA	SSL v3	0002	No
NULL_MD5	SSL v3	0001	No

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. Uma instrução TTLSEnvironmentAdvancedParms é associada ao `TTLSEnvironmentAction` pela propriedade **`TTLSEnvironmentAdvancedParmsRef`** ..

   Essa instrução pode ser usada para especificar quais protocolos SSL e TLS estão ativados. Com IBM MQ você deve ativar apenas o protocolo único que corresponde ao nome do conjunto de cifras usado na instrução `TTLSCipherParms` .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

O conjunto completo de instruções é o seguinte e deve ser aplicado ao agente de política:

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   SERVER
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Etapa 3: Remover SSLCIPH do z/OS canal

Remova o CipherSpec do canal z/OS usando o comando a seguir:

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

Etapa 4: Iniciar o canal

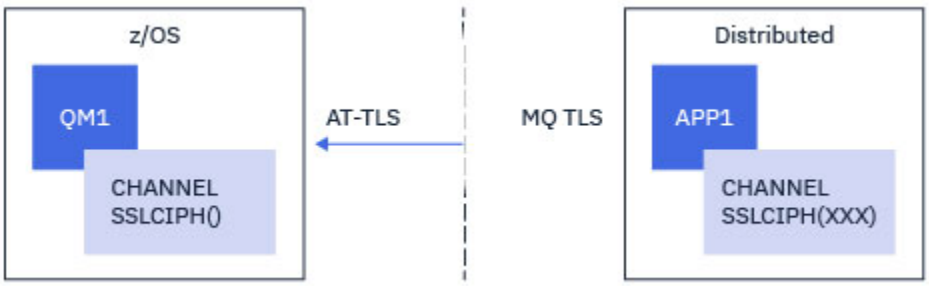
Uma vez que o canal tenha iniciado, ele estará usando uma combinação de AT-TLS e TLS do IBM MQ.

Atenção: As instruções AT-SLT anteriores são apenas uma configuração mínima Há outras Instruções de política AT-TLS com AT-TLS que não são documentadas aqui, e poderiam ser usadas com o IBM MQ dependendo da necessidade. No entanto, o IBM MQ foi testado apenas com as políticas descritas

z/OS *Configurando o AT-TLS em um canal de entrada para um Gerenciador de Filas IBM MQ for Multiplatforms usando um CipherSpecs de alias*

Como você configura o AT-TLS em um canal de entrada por meio de um Gerenciador de Filas IBM MQ for Multiplatforms para um Gerenciador de Filas IBM MQ for z/OS. Neste caso, o canal no Gerenciador de Filas z/OS é um canal receptor que não tem o conjunto de atributos SSLCIPH, e o canal no Gerenciador de Filas não z/OS é um canal emissor com o conjunto de atributos SSLCIPH para um CipherSpec de alias.

Neste exemplo, um par de canais emissor-receptor existente, que usa qualquer CipherSpec TLS 1.3, será ajustado para que o canal receptor utilize AT-TLS em vez de IBM MQ TLS.



Outros protocolos TLS e CipherSpecs podem ser usados fazendo pequenos ajustes na configuração. Outros tipos de canal de mensagens, além de canais emissores e receptores de cluster, poderiam ser usados sem alteração na configuração de AT-TLS.

Procedimento

Etapa 1: Pare o canal

Etapa 2: Crie e aplique uma política AT-TLS

É necessário criar as instruções AT-TLS a seguir para este cenário:

1. Uma instrução TTLSRule para corresponder conexões de entrada com o espaço de endereço do iniciador do canal por meio do endereço IP do canal emissor. Aqui, mais filtragem foi incluída para corresponder a um nome de tarefa do iniciador de canal específico.

```
TTLSRule                        REMOTE-TO-CSQ1
{
  LocalAddr                     ALL
  LocalPortRange                1414
  RemoteAddr                    123.456.78.9
  Jobname                       CSQ1CHIN
  Direction                     INBOUND
  TTLSGroupActionRef            CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef      CSQ1-INBOUND-ENVIRONMENT-ACTION
}
```

   As correspondências de regra anteriores com relação às conexões que vão para a tarefa CSQ1CHIN na porta local 1414 do endereço IP remoto 123.456.78.9.

   Mais opções avançadas de filtragem são descritas em TTLSRule.

2. Uma instrução TTLSGroupAction ativando a regra. O `TTLSRule` referencia o `TTLSGroupAction` usando a propriedade **`TTLSGroupActionRef`**.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Uma instrução TTLSEnvironmentAction está associada com o `TTLSRule` pela propriedade **`TTLSEnvironmentActionRef`**. Um `TTLSEnvironmentAction` configura o Ambiente TLS e especifica qual conjunto de chaves a utilizar.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

O AT-TLS fornece a capacidade de conceder autenticação mútua, que é o equivalente a utilizar o atributo de canal SSLCAUTH. Isso é feito por ter uma instrução `TTLSEnvironmentAction` com um valor **HandshakeRole** de *ServerWithClientAuth* para a instrução de entrada `TTLSEnvironmentAction`.

4. Uma instrução TTLSKeyringParms é associada ao `TTLSEnvironmentAction` pela propriedade **TTLSKeyringParmsRef** e define o conjunto de chaves usado por AT-TLS.

   O conjunto de chaves deve conter certificados confiáveis pelo Gerenciador de Filas remoto não z/OS. Este conjunto de chaves pode ser definido da mesma forma que um conjunto de chaves usado pelo inicializador de canal; consulte "Configuring your z/OS system to use TLS" na página 259.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Uma instrução TTLSCipherParms associada com o `TTLSEnvironmentAction` pela propriedade **TTLSCipherParmsRef**

   Esta instrução deve conter pelo menos um nome de conjunto de cifras que esteja incluído no CipherSpec de alias configurado no canal emissor remoto.

   Nota: Os nomes de conjunto de cifras AT-TLS não correspondem necessariamente aos nomes de CipherSpec do IBM MQ. No entanto, é possível localizar o nome do conjunto de cifras AT-TLS que corresponde a um nome de CipherSpec do IBM MQ localizando o nome de CipherSpec do IBM MQ na tabela a seguir e fazendo referência cruzada da coluna de código hexadecimal com a coluna de caracteres expandida da Tabela 2 no tópico de instrução TTLSCipherParms.

Tabela 86. CipherSpecs no z/OS do IBM MQ for z/OS 9.2.0

CipherSpec	Protocolo	Código Hexadecimal	Ativado por padrão
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sim
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sim
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sim
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sim
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sim
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sim
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sim
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sim
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sim
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sim
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sim

Tabela 86. CipherSpecs no z/OS do IBM MQ for z/OS 9.2.0 (continuação)

CipherSpec	Protocolo	Código Hexadecimal	Ativado por padrão
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sim
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sim
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	No
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	No
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	No
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	No
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	No
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	No
TRIPLE_DES_SHA_US	SSL v3	000A	No
RC4_SHA_US	SSL v3	0005	No
RC4_MD5_US	SSL v3	0004	No
DES_SHA_EXPORT	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	No
RC2_MD5_EXPORT	SSL v3	0006	No
NULL_SHA	SSL v3	0002	No
NULL_MD5	SSL v3	0001	No

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

Atenção: Se o gerenciador de filas e a política de AT-TLS suportarem TLS 1.3, somente os CipherSpecs de alias que contêm pelo menos um TLS 1.3 CipherSpec permitirão que o canal seja iniciado. Por exemplo, usar ANY_TLS12 resulta no canal falhando ao iniciar, mesmo se TTLSCipherParms contiver TLS 1.2 CipherSpecs, mas usando ANY_TLS12_OR_HIGHER ou ANY_TLS13 permite que o canal seja iniciado. Consulte “Relacionamento entre as configurações do CipherSpec de alias” na página 448 para uma explicação.

6. Uma instrução TTLSEnvironmentAdvancedParms é associada ao TTLSEnvironmentAction pela propriedade **TTLSEnvironmentAdvancedParmsRef** ..

   Essa instrução pode ser usada para especificar quais protocolos SSL e TLS estão ativados e deve ser consistente com os conjuntos de cifras na instrução TTLSCipherParms

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

O conjunto completo de instruções é o seguinte e deve ser aplicado ao agente de política:

```
TTLSRule                      REMOTE-TO-CSQ1
{
  LocalAddr                   ALL
  LocalPortRange              1414
  RemoteAddr                  123.456.78.9
  Jobname                     CSQ1CHIN
  Direction                   INBOUND
  TTLSGroupActionRef          CSQ1-GROUP-ACTION
  TTLSEnvironmentActionRef    CSQ1-INBOUND-ENVIRONMENT-ACTION
}

TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}

TTLSEnvironmentAction             CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                   SERVER
  TTLSKeyringParmsRef             CSQ1-KEYRING
  TTLSCipherParmsRef              CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef CSQ1-ENVIRONMENT-ADVANCED
}

TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}

TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256

}

TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Etapa 3: Remover SSLCIPH do z/OS canal

Remova o CipherSpec do canal z/OS usando o comando a seguir:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Etapa 4: Iniciar o canal

Uma vez que o canal tenha iniciado, ele estará usando uma combinação de AT-TLS e TLS do IBM MQ.

Atenção: As instruções AT-SLT anteriores são apenas uma configuração mínima Há outras Instruções de política AT-TLS com AT-TLS que não são documentadas aqui, e poderiam ser usadas

com o IBM MQ dependendo da necessidade. No entanto, o IBM MQ foi testado apenas com as políticas descritas

Reconfigurando as chaves secretas SSL e TLS

O IBM MQ suporta a reinicialização de chaves secretas em gerenciadores de filas e clientes.

As chaves secretas são reconfiguradas quando um número especificado de bytes criptografados de dados passa pelo canal. Se as pulsações de canal forem ativadas, a chave secreta será reconfigurada de os dados serem enviados ou recebidos após uma pulsação de canal.

O valor de reconfiguração de chave é sempre configurado pelo lado inicial do canal do IBM MQ.

Gerenciador de filas

Para um gerenciador de fila, use o comando **ALTER QMGR** com o parâmetro **SSLRKEYC** para configurar os valores usados durante a renegociação de chave

IBM i No IBM i, use o **CHGMQM** com o parâmetro **SSLRSTCNT**

Cliente MQI

Por padrão, os clientes MQI não renegociam a chave secreta. É possível fazer um cliente MQI renegociar a chave em qualquer uma das três formas. Na lista a seguir, os métodos são mostrados em ordem de prioridade. Se você especificar diversos valores, o valor de prioridade mais alto será usado.

1. Usando o campo KeyResetCount na estrutura MQSCO em uma chamada MQCONNX.
2. Usando a variável de ambiente **MQSSLRESET**..
3. Ao configurar o atributo **SSLKeyResetCount** na sub-rotina SSL do arquivo de configuração do cliente,

Essas variáveis podem ser configuradas para um número inteiro no intervalo de 0 a 999999999, representando o número de bytes não criptografados enviados e recebidos em uma conversa TLS antes de a chave secreta TLS ser renegociada. A especificação de um valor igual a 0 indica que as chaves secretas TLS nunca serão renegociadas. Se você especificar uma contagem de reconfiguração de chave secreta TLS no intervalo de 1 byte a 32 KB, os canais TLS usarão uma contagem de reconfiguração de chave secreta de 32 KB. Isto é para evitar reconfigurações de chave excessivas que ocorreriam para valores pequenos de reconfiguração de chave secreta TLS.

Se um valor maior que zero for especificado e as pulsações de canal forem ativadas para o canal, a chave secreta também será renegociada antes dos dados da mensagem serem enviados ou recebidos após uma pulsação de canal.

A contagem de bytes até a próxima renegociação de chave secreta é reconfigurada após cada renegociação bem-sucedida.

Java

Para o IBM MQ classes for Java, um aplicativo pode reconfigurar a chave secreta de uma das maneiras a seguir:

- Configurando o campo sslResetCount na classe MQEnvironment.
- Configurando uma propriedade de ambiente MQC.SSL_RESET_COUNT_PROPERTY em um objeto Hashtable. O aplicativo designa, então, a hashtable para o campo `properties` na classe MQEnvironment ou passa a hashtable para um objeto MQQueueManager em seu construtor.

Se o aplicativo usar mais de uma dessas maneiras, as regras usuais de precedência se aplicam. Consulte Classe com.ibm.mq.MQEnvironment para as regras de precedência.

O valor do campo sslResetCount ou a propriedade de ambiente MQC.SSL_RESET_COUNT_PROPERTY representa o número total de bytes enviados e recebidos pelo código do cliente IBM MQ classes for Java antes que a chave secreta seja renegociada. O número de bytes enviados é o número antes da criptografia

e o número de bytes recebidos é o número após a decriptografia. O número de bytes também inclui informações de controle enviadas e recebidas pelo cliente IBM MQ classes for Java.

Se a contagem de reconfiguração for zero, que é o valor padrão, a chave secreta nunca será renegociada. A contagem de reconfiguração será ignorada se nenhum CipherSuite for especificado.

JMS

Para o IBM MQ classes for JMS, a propriedade SSLRESETCOUNT representa o número total de bytes enviados e recebidos por uma conexão antes que chave secreta que é usada para criptografia seja renegociada. O número de bytes enviados é o número antes da criptografia e o número de bytes recebidos é o número após a decriptografia. O número de bytes também inclui informações de controle enviadas e recebidas pelo IBM MQ classes for JMS. Por exemplo, para configurar um objeto ConnectionFactory que possa ser usado para criar uma conexão por meio de um canal MQI ativado por TLS (Segurança da Camada de Transporte) com uma chave secreta que seja renegociada após a passagem de 4 MB de dados, emita o comando a seguir para JMSAdmin:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

Se o valor de SSLRESETCOUNT for zero, que é o valor padrão, a chave secreta nunca será renegociada. A propriedade SSLRESETCOUNT será ignorada se SSLCIPHERSUITE não estiver configurado.

.NET

Para .NET clientes não gerenciados, a propriedade de número inteiro **SSLKeyResetCount** indica o número de bytes não criptografados enviados e recebidos em uma conversa TLS antes que a chave secreta seja renegociada. Para obter mais informações sobre o uso de propriedades do objeto no IBM MQ classes for .NET, consulte Obtendo e configurando valores de atributo

Para clientes gerenciados .NET, a classe SSLStream não suporta reconfiguração/renegociação de chave secreta. No entanto, para ser consistente com outros clientes IBM MQ , o cliente IBM MQ gerenciado .NET permite que os aplicativos configure **SSLKeyResetCount**. Para obter mais informações, veja Reconfiguração ou renegociação de chave secreta.

XMS .NET

Para clientes não gerenciados XMS .NET, veja Conexões seguras com um gerenciador de filas do IBM MQ.

Referências relacionadas
ALTER QMGR
DISPLAYQMGR
Alterar Gerenciador da Fila de Mensagens (CHGMQM)
Exibir Gerenciador da Fila de Mensagens (DSPMQM)

Implementando confidencialidade em programas de saída do usuário

Implementando confidencialidade em saídas de segurança

As saídas de segurança podem exercer uma função no serviço de confidencialidade gerando e distribuindo a chave simétrica para criptografia e decriptografia de dados que flui no canal. Uma técnica comum aplicada utiliza a tecnologia PKI.

Uma saída de segurança gera um valor de dados aleatório, criptografa-os com a tecla pública do gerenciador ou usuário de fila que a saída do parceiro está representando e envia os dados criptografados a seu parceiro em uma mensagem de segurança. A saída de segurança do parceiro decriptografa os valores de dados aleatórios com a tecla privativa do gerenciador ou usuário de fila que os está representando. Cada saída de segurança pode agora utilizar o valor de dados aleatórios para derivar a

chave simétrica, independentemente de outros utilizando um algoritmo conhecido de ambos. De forma alternativa, podem usar o valor de dados aleatórios como a chave.

Se a primeira saída de segurança não autenticou seu parceiro até então, a mensagem de segurança seguinte enviada pelo parceiro poderá conter uma valor inesperado criptografado com a chave simétrica. A primeira saída de segurança pode então autenticar seu parceiro verificando se a saída de segurança dele conseguiu criptografar o valor esperado corretamente.

As saídas de segurança podem também usar esta oportunidade para concordar o algoritmo para criptografia e decriptografia de dados que fluem no canal, se mais de um algoritmo estiver disponível para uso.

Implementando confidencialidade em saídas de mensagem

Uma saída de mensagem na extremidade de envio de um canal pode criptografar os dados do aplicativo em uma mensagem e outra saída na extremidade de recepção do canal pode descriptografar os dados. Por razões de desempenho, um algoritmo de tecla simétrica é normalmente utilizado para este propósito. Para obter mais informações sobre como a chave simétrica pode ser gerada e distribuída, consulte “Implementando confidencialidade em programas de saída do usuário” na página 475.

Os cabeçalhos de uma mensagem, como da fila de transmissão, MQXQH, que inclui o descritor da mensagem interna, não devem ser criptografados por uma saída de mensagem. Isso porque a conversão de dados dos cabeçalhos da mensagem ocorre depois que uma saída de mensagem é chamada na extremidade de envio ou antes de ser chamada na extremidade de recepção. Se os cabeçalhos forem criptografados, a conversão de dados falhará e o canal parará.

Implementando confidencialidade em saídas de envio e de recebimento

As saídas de envio e recebimento podem ser utilizadas para criptografar e descriptografar dados que passam por um canal. Elas são mais apropriadas que as saídas de mensagens que fornecem esse serviço pelos seguintes motivos:

- Em um canal de mensagem, os cabeçalhos da mensagem podem ser criptografados assim como os dados do aplicativo nas mensagens.
- As saídas de envio e recebimento podem ser usadas em canais MQI assim como em canais de mensagens. Os parâmetros nas chamadas MQI podem não conter dados sensíveis do aplicativo que tenham que ser protegidos enquanto passam em um canal MQI. Portanto, é possível usar as mesmas saídas de envio e recebimento nos dois tipos de canais.

A confidencialidade na saída da API e saída cruzada da API

Os dados do aplicativo em uma mensagem podem ser criptografados por uma saída API ou saída cruzada da API quando a mensagem for colocada pelo aplicativo de envio e decriptografada por uma segunda saída quando a mensagem for recuperada pelo aplicativo de recebimento. Por razões de desempenho, um algoritmo de chave simétrica é normalmente usado para este propósito. No entanto, ao nível do aplicativo, em que muitos usuários podem estar enviando mensagens uns aos outros, o problema é como assegurar que somente o receptor pretendido de uma mensagem seja capaz de decriptografá-la. Uma solução é utilizar uma chave simétrica diferente para cada par de usuários que enviem mensagens uns aos outros. Mas essa solução pode ser difícil e demorada para administrar, particularmente se os usuários pertencerem a organizações diferentes. Um modo padrão de resolver este problema é conhecido como *envelopamento digital* e utiliza tecnologia PKI.

Quando um aplicativo coloca uma mensagem em uma fila, uma saída API ou saída cruzada da API gera uma chave simétrica aleatória e usa a chave para criptografar os dados do aplicativo na mensagem. A saída criptografa a chave simétrica com a chave pública do receptor desejado. Então, ela substitui os dados do aplicativo na mensagem pelos dados criptografados do aplicativo e a chave simétrica criptografada. Deste modo, somente o receptor pretendido poderá decriptografar a chave simétrica e, portanto, os dados do aplicativo. Se uma mensagem criptografada tiver mais de um possível receptor desejado, a saída poderá criptografar uma cópia da chave simétrica para cada receptor desejado.

Se diferentes algoritmos para criptografar e decriptografar dados do aplicativo estiverem disponíveis para uso, a saída poderá incluir o nome do algoritmo que foi usado.

z/OS Confidentiality for data at rest on IBM MQ for z/OS with data set encryption

IBM MQ for z/OS can harden customer and configuration data by writing the data to the active log data sets, the archive log data sets, page sets, boot strap data sets (BSDS), and shared message data sets (SMDS).

z/OS provides efficient, policy-based encryption of data sets. IBM MQ for z/OS supports z/OS data set encryption for:

- Active log data sets; see note “1” on page 477
- Archive log data sets; see note “2” on page 477
- Page sets; see note “1” on page 477
- BSDS; see note “2” on page 477
- CSQINP* data sets; see note “2” on page 477
- SMDS; see note “1” on page 477

This provides confidentiality of data at rest on an individual z/OS queue manager.

Notes:

1. From IBM MQ for z/OS 9.2.0, z/OS data set encryption for active logs. page sets, and SMDS are supported.
2. Data set encryption for archive logs, BSDS and CSQINP* data sets is supported on all versions of IBM MQ for z/OS.
3. IBM MQ Advanced Message Security provides an alternative mechanism of protecting data at rest. In addition AMS also protects data in memory and in flight

See Using the z/OS data set encryption enhancements for more information about z/OS data set encryption.

Configuration of z/OS data set encryption is outside of the control of IBM MQ for z/OS. Encryption settings take effect when the data set is created.

This means that any existing data sets need to be recreated before a new data set encryption policy can be used.

IBM MQ for z/OS can run with a mixture of encrypted and non-encrypted data sets, but a standard configuration would encrypt all, or none, of the data sets used.

z/OS Overview of steps to encrypt an IBM MQ for z/OS data set

How you encrypt an IBM MQ for z/OS data set.

Before you begin

You must ensure that you have configured z/OS data set encryption correctly in your enterprise. If you are setting up data set encryption in a queue sharing group, you must configure z/OS data set encryption for data sharing.

Note: A z/OS encrypted data set must be an extended format data set.

Procedure

1. Set up encryption key and `key-label` in RACF to use to encrypt the data set.
2. Create a profile for `key-label` in the RACF CSFKEYS class.

3. Grant READ access to the user Id of the queue manager, and any other user Ids that need access to the encrypted data.

   This might include user IDs that are used to run print utilities against the data set. For example, the user running CSQUTIL SCOPY would need to decrypt the relevant page set.
4. Associate the encryption `key-label` with the data set name.

   You can do this by using an SMS data class, or a RACF DFP segment, for the data set name or high-level qualifier.

   You can also associate the `key-label` with the data set when the data set is allocated.
5. Rename any existing data set using IDCAMS ALTER.
6. Re-allocate the data set with the appropriate attributes.
7. Copy the contents of the renamed data set to the new data set using IDCAMS REPRO.

   The data is encrypted by the action of copying it into the data set.
8. Repeat steps “4” on page 478 to “6” on page 478 for any other data sets that need to be encrypted.

z/OS Example of how to encrypt queue manager active logs

The following topics guide you through the process of enabling data set encryption on existing active logs.

Note: The process for other data sets is similar to that for active logs.

In this example:

- Queue manager CSQ1 is run under user QMCSQ1, and has active log data sets CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, and so on
- The hardware and software environment is capable of using z/OS data set encryption
- RACF is used as the SAF
- The queue manager has been stopped

Carry out the procedure in the following order:

1. “Configuring the data set encryption key for the queue manager” on page 478
2. “Configuring data set encryption for the log data sets” on page 479

z/OS Configuring the data set encryption key for the queue manager

How you configure a data set encryption key for a queue manager.

About this task

This task is a prerequisite for “Configuring data set encryption for the log data sets” on page 479.

Procedure

1. Set up an AES-256 bit encryption DATA key with a label, for example, CSQ1DSKY, using the z/OS key generator utility program (KGUP).
2. Define the RACF CSFKEYS profile for the CSQ1DSKY encryption key, by issuing the following command:

   ```
   RDEFINE CSFKEYS CSQ1DSKY UACC(NONE)
   ```
3. Configure the ICSF segment of the profile to allow the key to be used as a protected key, by issuing the following command:

   ```
   RALTER CSFKEYS CSQ1DSKY ICSF(SYMCPACFWRAP(YES) SYMCPACFRET(YES))
   ```
4. Allow the queue manager to use the encryption key by giving QMCSQ1 READ access to the profile, by issuing the following command:

   ```
   PERMIT CSQ1DSKY CLASS(CSFKEYS) ID(QMCSQ1) ACCESS(READ)
   ```

Give the same access to any administrative user that needs to read or write the encrypted data set.

5. Refresh the CSFKEYS class by issuing the following command.

```
SETROPTS RACLIST(CSFKEYS) REFRESH
```

What to do next

Configure data set encryption for the data sets as described in “Configuring data set encryption for the log data sets” on page 479

z/OS Configuring data set encryption for the log data sets

How you configure the encryption on the log data sets.

Before you begin

Ensure that you have read:

Overview of steps to encrypt an IBM MQ for z/OS data set, and carried out the procedure in “Configuring the data set encryption key for the queue manager” on page 478

About this task

This method uses the DFP segment of a RACF generic profile, so that you can use the encryption key for all new data sets that match the profile.

Alternatively, you can configure and use an SMS data class, or the key label can be specified directly when allocating the data set.

As previously described, in this example, queue manager CSQ1 is run under user QMCSQ1, and has active log data sets CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, and so on.

Procedure

1. Create the generic profile if it does not exist, by issuing the following command:

```
ADDSD 'CSQ1.LOGS.*' UACC(NONE)
```

2. Permit the queue manager user alter access on the profile, by issuing the following command:

```
PERMIT 'CSQ1.LOGS.*' ID(QMCSQ1) ACCESS(ALTER)
```

   Also, permit the appropriate access needed for any administrative user.

3. Add the DFP segment with the encryption key label by issuing the following command:

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

   Note: You must use the same encryption key that you used in configuring the data set encryption key for the queue manager.

4. Refresh the generic dataset profiles by issuing the following command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. Rename each log data set to a backup, then recreate and restore the data, using IDCAMS. The following JCL fragment converts CSQ1.LOGS.LOGCOPY1.DS001:

   a) Rename the data set to a back-up

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME DATASET TO BACKUP                                    */
  /*-------------------------------------------------------------*/
```

```
  ALTER 'CSQ1.LOGS.LOGCOPY1.DS001'                    -
        NEWNAME('CSQ1.BAK.LOGS.LOGCOPY1.DS001')
```

b) Redefine the data set.

The new data set will be encrypted due to the RACF profile.

Note: Replace ++EXTDCLASS++ with the name of the extended format data class you want to use for the data set.

```
//REDEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* REDEFINE THE DATASET                                        */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                    -
         (NAME(CSQ1.LOGS.LOGCOPY1.DS001)            -
         LINEAR                                     -
         SHAREOPTIONS(2 3)                          -
         MODEL(CSQ1.BAK.LOGS.LOGCOPY1.DS001)        -
         DATACLAS(++EXTDCLASS++))
```

c) Copy the data from the backup into the recreated data set.

This step encrypts the data:

```
//RESTORE  EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RESTORE DATA INTO ENCRYPTED LOG                             */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)     -
        OUTDATASET(CSQ1.LOGS.LOGCOPY1.DS001)
```

What to do next

Repeat Step “5” on page 479 for all active log data sets.

Only a single encryption key is required, and all data sets can be associated with the same key label.

Restart queue manager CSQ1. Use the output from the DISPLAY LOG command to verify that the log data sets have been encrypted.

z/OS Considerations for z/OS data set encryption in a queue sharing group

Each queue manager in a queue sharing group (QSG) must be able to read the logs, BSDS, and shared message data sets (SMDS), of every other queue manager in the QSG.

This means that each system on which a member of the QSG can run, must meet the requirements for z/OS data set encryption, and all the key labels and encryption keys used to protect the data sets for each queue manager in the QSG must be available on each system.

A queue manager prior to IBM MQ for z/OS 9.1.4 cannot access an encrypted active log data set.

A queue manager prior to IBM MQ for z/OS 9.1.5 cannot access an encrypted SMDS.

Before making use of z/OS data set encryption, you should migrate all queue managers in a QSG to at least IBM MQ for z/OS 9.1.5.

If a queue manager in a QSG is started with any encrypted active log data set, and any other queue manager in the QSG has been started, but was not last started with a version of IBM MQ for z/OS that supports encrypted active logs, the queue manager with the encrypted active log terminates abnormally with abend code 5C6-00F50033.

You can convert a QSG to use encrypted active logs and SMDS without a full outage, by:

1. Migrating each queue manager to at least IBM MQ for z/OS 9.1.5 in turn.
2. Converting active logs to encrypted data sets for each queue manager in turn. This requires the queue manager to be shut down and then restarted.

   At the same time, it is likely that page sets and archive logs would be enabled for encrypted data sets too, but this does not affect QSG migration.

   The procedure for converting each data set is described in "Example of how to encrypt queue manager active logs" on page 478
3. Converting SMDS to encrypted data sets for each individual CF structure in turn by:
   a. Issuing the command RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name) to suspend queue manager access to the SMDS.

      Note that during this time, the data on the shared queues associated with the SMDS is temporarily unavailable.
   b. Converting each data set that makes up the SMDS to encrypted data sets, using the procedure described in "Example of how to encrypt queue manager active logs" on page 478.
   c. Issuing the command RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name) to resume queue manager access to the SMDS.

Attention: You should shut the queue manager down cleanly prior to converting the logs, and coupling facility structure recovery might not be possible during the conversion, as the active log data sets will be temporarily unavailable.

z/OS Backwards migration considerations when using z/OS data set encryption

You need to consider the following when backwards migrating a queue manager, which has one or more encrypted data sets.

z/OS data set encryption is supported on the following IBM MQ for z/OS data sets:

- Active log data sets
- Archive log data sets
- Page sets
- BSDS
- SMDS
- CSQINP* data sets

There are no backwards migration considerations for BSDS, archive log, or CSINP* data sets.

However, there are considerations for

- SMDS
- Page set and
- Active log

data sets, as using these with z/OS data set encryption is not supported in IBM MQ for z/OS 9.1.0, and earlier, long term support releases.

Prior to backwards migration, all encryption policies for SMDS, page set, and active log data sets need to be removed and the data decrypted. This process is described in "Removing data set encryption from a data set" on page 482.

Attention: If the queue manager to be backwards migrated is part of a queue sharing group (QSG), read the "Queue sharing group considerations" on page 483 section first.

Removing data set encryption from a data set

This example describes how to remove data set encryption from the log data set CSQ1.LOGS.LOGCOPY1.DS001. You can use an equivalent process for SMDS and page sets.

The example assumes that:

- RACF is the SAF
- The queue manager that uses the data set has been stopped
- The encryption key label has been associated with the generic RACF profile CSQ1.LOGS.*

Carry out the following procedure:

1. Copy the data from the data set to a back-up data set.
   a. Define a backup data set which is not associated with an encryption key label.

      Note: Replace ++EXTDCLASS++ with the name of the extended format data class you want to use for the data set.

```
//DEFINE EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* DEFINE UNENCRYPTED DATA SET                                 */
  /*-------------------------------------------------------------*/
  DEFINE CLUSTER                                        -
         (NAME(CSQ1.BAK.LOGS.LOGCOPY1.DS001)         -
         LINEAR                                       -
         SHAREOPTIONS(2 3)                            -
         MODEL(CSQ1.LOGS.LOGCOPY1.DS001)              -
         DATACLAS(++EXTDCLASS++))
/*
```

   b. Copy the data from the original data set to the backup. This step decrypts the data.

```
//COPY     EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* COPY DATA INTO UNENCRYPTED DATA SET                         */
  /*-------------------------------------------------------------*/
  REPRO INDATASET(CSQ1.LOGS.LOGCOPY1.DS001)           -
        OUTDATASET(CSQ1.BAK.LOGS.LOGCOPY1.DS001)
/*
```

   c. Delete the original data set

```
//DELETE   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* DELETE ORIGINAL                                             */
  /*-------------------------------------------------------------*/
  DELETE ('CSQ1.LOGS.LOGCOPY1.DS001')
/*
```

   d. Rename the backup to the original data set name. The data remains unencrypted

```
//RENAME   EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------------*/
  /* RENAME UNENCRYPTED DATA SET                                 */
  /*-------------------------------------------------------------*/
  ALTER CSQ1.BAK.LOGS.LOGCOPY1.DS001'                -
         NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001)
  ALTER 'CSQ1.BAK.LOGS.LOGCOPY1.DS001.*'              -
         NEWNAME('CSQ1.LOGS.LOGCOPY1.DS001.*')
/*
```

2. Optionally, repeat this process for other data sets that have an encryption key label associated with them through the CSQ1.LOGS.* generic profile.
3. Optionally, if all data sets associated with the CSQ1.LOGS.* generic profile have been decrypted, remove the DATAKEY associated with the generic profile by issuing the following command

```
ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
```

4. Refresh the generic dataset profiles by issuing the following command:

```
SETROPTS GENERIC(DATASET) REFRESH
```

5. Restart the queue manager.
6. If the encryption key is no longer needed, delete it, and delete its associated RACF profile from the CSFKEYS class.

Queue sharing group considerations

If a queue manager that is part of a queue sharing group is going to be backwards migrated to a version of IBM MQ for z/OS that does not support data set encryption then all of the active log data sets and SMDS of all queue managers in the QSG need to have their data set encryption policies removed, and their data decrypted.

This applies regardless of whether a single member of QSG is backwards migrated, or all members of the QSG.

You can achieve removal of encryption policies, and decryption of data, without a full QSG outage by:

1. Shutting down each queue manager in the QSG in turn, removing the encryption policies and decrypting the data from its active logs, using the process described in “Removing data set encryption from a data set” on page 482.

   If the queue manager is to be backwards migrated, its page set should also be decrypted at this time. Then restart the queue manager.
2. Removing the encryption policies and decrypting the data for the SMDS of each individual CF structure in turn by:
   a. Issuing the command

   ```
   RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name)
   ```

   to suspend queue manager access to the SMDS. During this time the data on the shared queues associated with the SMDS will be temporarily unavailable.
   b. Following the process in “Removing data set encryption from a data set” on page 482 for each data set which makes up the SMDS.
   c. Issuing the command

   ```
   RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name)
   ```

   to resume queue manager access to the SMDS.

Using z/OS data set encryption with a queue manager that does not support it

If you accidentally backwards migrate a queue manager to a version of IBM MQ for z/OS that does not support data set encryption, and forget to remove the encryption policies and decrypt the data you get an error when the queue manager tries to access the data set.

The error depends on the data set type and is shown in the following table.

Note: If one or more of these errors occur, you need to follow the processes described in “Removing data set encryption from a data set” on page 482 for the affected data set. These can be performed without changing the version of IBM MQ for z/OS.

Data set	Error if queue manager does not support z/OS data set encryption
Page set 0	Abend 5C6-00C91400 at queue manager start
Page sets 1-99	MQRC 2193 "Pageset error" when accessing page set, for example, on MQPUT
Active log	Abend 5C6-00E80084 at queue manager start
SMDS	Message IEC161I-122 logged "The data set has a KEYLABEL, but the user did not specify that the application could handle encryption". SMDS marked AVAIL(ERROR).

Integridade de dados de mensagens

Para manter a integridade dos dados é possível usar vários tipos de programa de saída de usuário para fornecer trechos de mensagens ou assinaturas digitais para suas mensagens.

Integridade de dados

Implementando integridade de dados em mensagens

Ao usar TLS, sua opção de CipherSpec determina o nível de integridade de dados na empresa. Se você usar o IBM MQ Advanced Message Service (AMS), é possível especificar a integridade para uma mensagem exclusiva.

Implementando integridade de dados em saídas de mensagem

Uma mensagem pode ser assinada digitalmente por uma saída de usuário na extremidade de envio de um canal. A assinatura digital pode então ser verificada por uma saída de mensagem na extremidade de recepção de um canal para detectar se a mensagem foi modificada deliberadamente.

Alguma proteção pode ser proporcionada utilizando-se uma compilação de mensagens ao invés de uma assinatura digital. Uma compilação de mensagens pode ser efetiva contra violação casual ou indiscriminada, mas não evita que pessoas mais informadas alterem ou substituam a mensagem e gerem uma compilação completamente nova. Isto é particularmente verdadeiro quando o algoritmo utilizado para gerar a compilação da mensagem é bem conhecido.

Implementando integridade de dados em saídas de envio e recebimento

Em um canal de mensagem, as saídas de mensagem são mais apropriadas para fornecer esse serviço porque uma saída pode acessar uma mensagem inteira. Em um canal MQI, os parâmetros nas chamadas MQI podem conter dados do aplicativo que precisem de proteção e somente as saídas de envio e recebimento podem fornecer essa proteção.

Implementando integridade de dados na saída da API ou saída cruzada da API

Uma mensagem pode ser assinada digitalmente por uma saída API ou saída cruzada da API ao ser colocada pelo aplicativo de envio. A assinatura digital pode então ser verificada por uma segunda saída quando a mensagem for recuperada pelo aplicativo receptor para detectar se a mensagem foi modificada deliberadamente.

Alguma proteção pode ser proporcionada utilizando-se uma compilação de mensagens ao invés de uma assinatura digital. Uma compilação de mensagens pode ser efetiva contra violação casual ou indiscriminada, mas não evita que pessoas mais informadas alterem ou substituam a mensagem e gerem uma compilação completamente nova. Isso é particularmente verdadeiro se o algoritmo que é usado para gerar o trecho da mensagem é conhecido,

Outras informações

Consulte a seção sobre “Ativando CipherSpecs” na página 427 para obter mais informações sobre como assegurar a integridade dos dados.

Tarefas relacionadas

Conectando dois gerenciadores de filas usando TLS

Conectando um Cliente a um Gerenciador de Filas de Forma Segura

Auditoria

É possível procurar por intrusões de segurança ou tentativas de intrusão usando mensagens do evento. Também é possível verificar a segurança do seu sistema usando o IBM MQ Explorer.

Para detectar tentativas de executar ações não autorizadas como conectar-se a um gerenciador de filas ou colocar uma mensagem em uma fila, examine as mensagens de eventos produzidas por seus gerenciadores de filas, sobretudo mensagens de eventos de autoridade. Para obter mais informações sobre mensagens de eventos do gerenciador de filas, consulte Eventos de filas do gerenciador e para obter mais informações sobre o monitoramento de eventos em geral, consulte Monitoramento de eventos.

Mantendo Clusters Seguros

Autorize ou evite que os gerenciadores de filas juntem os clusters ou coloquem as mensagens nas filas de cluster. Force um gerenciador de filas para deixar um cluster. Leve em conta algumas considerações adicionais ao configurar o TLS para clusters.

Parando o envio de mensagens por gerenciadores de filas desautorizados

Evite que os gerenciadores de filas desautorizados enviem mensagens para seu gerenciador de filas usando uma saída de segurança do canal.

Antes de começar

O armazenamento em cluster não tem efeito na maneira como as saídas de segurança funcionam. Você pode restringir o acesso a um gerenciador de filas da mesma maneira que faria em um ambiente de enfileiramento distribuído.

Sobre esta tarefa

Evite que os gerentes de filas selecionados enviem mensagens ao seu gerenciador de filas:

Procedimento

1. Defina um programa de saída de segurança do canal na definição de canal CLUSRCVR.
2. Grave um programa que autentica gerenciadores de filas que estão tentando enviar mensagens em seu canal do receptor de clusters e nega-lhes o acesso se não estiverem autorizados.

Como proceder a seguir

Os programas de saída de segurança do canal são chamados na inicialização e na rescisão de MCA.

Parando a Colocação de Mensagens de Gerenciadores de Filas Desautorizados em suas Filas

Use o canal para colocar o atributo de autoridade no canal do receptor de clusters para parar os gerenciadores de filas não autorizados de colocar mensagens nas suas filas. Autorize um gerenciador de filas remotas verificando o ID do usuário na mensagem usando RACF em z/OSou o OAM em Multiplataformas.

Sobre esta tarefa

Use as instalações de segurança de uma plataforma e o mecanismo de controle de acesso no IBM MQ para controlar o acesso às filas.

Procedimento

1. Para evitar que determinados gerenciadores de filas coloquem mensagens em uma fila, use os recursos de segurança disponíveis em sua plataforma.

   Por exemplo:

   - z/OS O RACF ou outros gerenciadores de segurança externa no IBM MQ for z/OS
   - Multi O OAM (Object Authority Manager) em outras Multiplataformas.
2. Use a autoridade put, PUTAUT, o atributo CLUSRCVR na definição de canal.

   O atributo PUTAUT permite que você especifique quais os identificadores de usuário devem ser usados para estabelecer a autoridade para colocar uma mensagem em uma fila.

   As opções no atributo PUTAUT são:

   DEF
   Use o ID do usuário padrão.

   z/OS No z/OS, a verificação pode envolver usar ambos o ID do usuário da rede o derivado de MCAUSER.

   CTX
   Use o ID do usuário nas informações de contexto associadas à mensagem.

   z/OS No z/OS a verificação pode envolver o uso do ID do usuário recebido da rede ou daquele derivado de MCAUSER ou ambos. Use esta opção se o link for confiável e autenticado.

   z/OS **ONLYMCA (somente z/OS)**
   Igual a DEF, mas qualquer ID do usuário recebido da rede não é usado. Use esta opção se o link não for confiável. Você deseja permitir somente um conjunto específico de ações nele, que são definidos para o MCAUSER.

   z/OS **ALTMCA (somente z/OS)**
   Como para CTX, mas qualquer ID do usuário recebido da rede não é usado.

Autorizando a Colocação de Mensagens em Filas de Cluster Remotas

On z/OS configure a autorização para colocar em uma fila de clusters usando o RACF Em Multiplataformas, autorize o acesso para conectar-se aos gerenciadores de fila e para colocar nas filas nesses gerenciadores de filas

Sobre esta tarefa

O comportamento padrão é realizar o controle de acesso no SYSTEM.CLUSTER.TRANSMIT.QUEUE. Observe que esse comportamento se aplica mesmo se você estiver usando diversas filas de transmissão.

O comportamento específico descrito neste tópico se aplica apenas quando você tiver configurado o atributo **ClusterQueueAccessControl** no arquivo qm.ini para ser *RQMNome*, conforme descrito no tópico Sub-rotina de segurança, e tiver reiniciado o gerenciador de filas.

Procedimento

- z/OS
  Para z/OS, emita os seguintes comandos:

```
RDEFINE MQQUEUE QMgrName.QUEUE. QueueName UACC(NONE)
PERMIT QMgrName.QUEUE. QueueName CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

- ALW

  Para sistemas AIX, Linux, and Windows, emita os comandos a seguir:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
  ```

- IBM i

  Para IBM i, emita os seguintes comandos:

  ```
  GRTMQMAUT OBJ(' QMgrName ') OBJTYPE(*MQM) USER(GroupName) AUT(*CONNECT)
  GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
  ```

  O usuário pode colocar mensagens apenas na fila de clusters especificada, e em nenhuma outra fila de clusters.

  Os nomes das variáveis possuem os seguintes significados:

  QMgrName
  O nome do gerenciador de filas. Em z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.

  GroupName
  O nome do grupo que receberá acesso.

  QueueName
  Nome da fila ou perfil genérico para o qual mudar as autorizações.

Como proceder a seguir

Se você especificar uma fila de resposta quando colocar uma mensagem em uma fila de clusters, o aplicativo consumidor deverá ter autoridade para enviar a resposta. Configure essa autoridade seguindo as instruções em "Concedendo autoridade para colocar mensagens em uma fila do cluster remoto" na página 402.

Conceitos relacionados

Sub-rotina de segurança no qm.ini

Impedindo que Gerenciadores de Filas se Juntem a um Cluster

Se um gerenciador de filas nocivo se unir a um cluster é difícil evitar o recebimento de mensagens que você não deseja que ele receba.

Procedimento

Se você desejar assegurar que somente determinados gerenciadores de filas autorizados se juntar a um cluster, você tem a opção de três técnicas:

- Usar registros de autenticação de canal, é possível bloquear a conexão do canal do cluster com base em: o endereço IP remoto, o nome do gerenciador de filas remotas ou o Nome distinto TLS fornecido pelo sistema remoto.
- Gravar um programa de saída para evitar que gerenciadores de filas não autorizados gravem em SYSTEM.CLUSTER.COMMAND.QUEUE. Não restrinja o acesso a SYSTEM.CLUSTER.COMMAND.QUEUE de forma que nenhum gerenciador de filas possa gravar nele ou você impediria que qualquer gerenciador de filas se junte ao cluster.
- Um programa de saída de segurança na definição de canal CLUSRCVR.

Saídas de segurança nos canais de cluster

Considerações extra ao usar saídas de segurança em canais de cluster.

Sobre esta tarefa

Quando um canal do emissor de clusters é iniciado pela primeira vez, ele usa atributos definidos manualmente por um administrador do sistema. Quando o canal é interrompido e reiniciado, ele seleciona os atributos da definição do canal do receptor de clusters correspondente. A definição de canal do emissor de clusters original é sobrescrita com os novos atributos, incluindo o atributo `SecurityExit`.

Procedimento

1. Deve-se definir uma saída de segurança em ambas as extremidades do emissor de cluster e o receptor de cluster de um canal.

   A conexão inicial deve ser feita com um handshake de saída de segurança, mesmo que o nome de saída de segurança seja enviado a partir da definição do receptor de cluster.
2. Valide o `PartnerName` na estrutura `MQCXP` na saída de segurança.

   A saída deve permitir que o canal inicie somente se o gerenciador de filas do parceiro estiver autorizado
3. Projete a saída de segurança na definição do receptor de cluster para ser iniciada pelo destinatário.
4. Se você projetá-la como iniciada pelo emissor, um gerenciador de filas não autorizado sem uma saída de segurança poderá unir-se ao cluster porque nenhuma verificação de segurança será executada.

   Não até o canal ser interrompido e reiniciado pode o nome SCYEXIT ser enviado a partir da definição do receptor de cluster e verificações de segurança integral feita.
5. Para visualizar a definição de canal do emissor de clusters que está atualmente em uso, use o comando:

   ```
   DISPLAY CLUSQMGR( queue manager ) ALL
   ```

   O comando exibe os atributos que foram enviados a partir da definição do receptor de clusters.
6. Para visualizar a definição original, use o comando:

   ```
   DISPLAY CHANNEL( channel name ) ALL
   ```
7. Você pode precisar definir uma saída de autodefinição de canal `CHADEXIT`, no gerenciador de filas do emissor de cluster, se os gerenciadores de filas estiverem em plataformas diferentes.

   Use a saída de definição automática de canal para configurar o atributo `SecurityExit` para um formato apropriado para a plataforma de destino.
8. Implementar e configurar a saída de segurança.

   z/OS

   O módulo de carregamento de saída de segurança deve estar no conjunto de dados especificado na instrução `CSQXLIB DD` do procedimento de espaço de endereço do inicializador de canais.

   Sistemas AIX, Linux, and Windows

   - A biblioteca de vínculo dinâmico de saída de segurança deve estar no caminho especificado no atributo SCYEXIT da definição de canal.
   - A biblioteca de vínculo dinâmico de saída de autodefinição de canal deve estar no caminho especificado no atributo `CHADEXIT` da definição do gerenciador de filas.

Forçando Gerenciadores de Filas Indesejáveis a Sair de um Cluster

Forçar um gerenciador de filas indesejado para deixar um cluster, emitindo o comando RESET CLUSTER em um gerenciador de filas de repositório completo.

Sobre esta tarefa

É possível forçar um gerenciador de filas indesejado para deixar um cluster. Se, por exemplo, um gerenciador de filas for excluído mas seus canais do receptor de clusters ainda estiverem definidos no cluster. Você pode desejar organizar.

Somente gerenciadores de fila de repositório completo têm autorização para ejetar um gerenciador de filas de um cluster.

Nota: Embora usar o comando RESET CLUSTER coercivamente remove um gerenciador de filas de um cluster, o uso de RESET CLUSTER por si só não impede que o gerenciador de filas se una ao cluster posteriormente. Para assegurar que o gerenciador de filas não se unirá novamente ao cluster, siga as etapas detalhadas em "Impedindo que Gerenciadores de Filas se Juntem a um Cluster" na página 487.

Siga este procedimento para ejetar o gerenciador de filas OSLO do cluster NORWAY:

Procedimento

1. Em um gerenciador de filas de repositório completo, emita o comando:

```
RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
```

2. Como alternativa use o QMID em vez de QMNAME no comando:

```
RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
```

   Nota: QMID é uma sequência, portanto, o valor de qmid deve ser colocado entre aspas simples, por exemplo, QMID('FR01_2019-07-15_14.42.42').

Resultados

O gerenciador de filas que é removido à força não muda; suas definições de cluster local mostram que ele está no cluster. As definições em todos os outros gerenciadores de filas não mostram isso no cluster.

Impedindo que gerenciadores de filas recebam mensagens

É possível evitar que um gerenciador de filas do cluster receba mensagens que ele não está autorizado a receber, usando programas de saída.

Sobre esta tarefa

É difícil parar a definição de uma fila por um gerenciador de filas que é membro de um cluster. Há um risco de que um gerenciador de filas nocivo se una a um cluster e defina sua própria instância de uma das filas no cluster. Agora ele pode receber mensagens que ele não está autorizado a receber. Para impedir que um gerenciador de filas receba mensagens, use uma das seguintes opções determinado no procedimento.

Procedimento

- Um programa de saída do canal em cada canal do emissor de clusters. O programa de saída usa o nome de conexão para determinar a adequação do gerenciador de filas de destino para que as mensagens sejam enviadas.
- Um programa de saída de carga de do cluster que usa os registros de destino para determinar a adequação da fila de destino e do gerenciador de filas para ser enviadas as mensagens.

SSL/TLS e clusters

Ao configurar o TLS para clusters, esteja ciente que uma definição de canal CLUSRCVR é propagada para outros gerenciadores de filas como um canal CLUSSDR autodefinido. Se um canal CLUSRCVR usar TLS, você deverá configurar o TLS em todos os gerenciadores de filas que se comunicarem usando o canal.

Para obter informações adicionais sobre o TLS, consulte “Protocolos de segurança TLS no IBM MQ” na página 25. O conselho lá geralmente é aplicável aos canais do cluster, mas talvez você deseje fornecer alguma consideração especial ao seguinte:

Em um cluster do IBM MQ uma determinada definição de canal CLUSRCVR é frequentemente propagada a muitos outros gerenciadores de filas nos quais é transformada em um CLUSSDR definido automaticamente. Subsequentemente o CLUSSDR definido automaticamente é usado para iniciar um canal para o CLUSRCVR. Se o CLUSRCVR for configurado para conectividade TLS, as considerações a seguir se aplicarão:

- Todos os gerenciadores de filas que desejam se comunicar com este CLUSRCVR devem ter acesso ao suporte de TLS. Esta provisão de TLS deve suportar CipherSpec para o canal.
- Os diferentes gerenciadores de filas para os quais os canais do emissor de clusters definidos automaticamente foram propagados terão, cada, um nome distinto diferente associado. Se a verificação de peer de nome distinto deve ser usada no CLUSRCVR ela deve ser configurada para que todos os nomes distintos que serão recebidos sejam correspondidos com sucesso.

  Por exemplo, vamos assumir que todos os gerenciadores de filas que hospedarão os canais do emissor de clusters que se conectarão a um determinado CLUSRCVR possuem certificados associados. Também vamos assumir que os nomes distintos em todos estes certificados definam o país como UK, a organização como IBM, a unidade da organização como IBM MQ Development e que todos possuam nomes comuns no formato DEVT.QMnnn, em que nnn é numérico.

  Neste caso, um valor SSLPEER de `C=UK, O=IBM, OU=IBM MQ Development, CN=DEVT.QM*` no CLUSRCVR permitirá que todos os canais do receptor de clusters necessários sejam conectados com sucesso, mas evitará a conexão de canais do receptor de clusters não desejados.
- Se cadeias de CipherSpec customizadas forem usadas, esteja ciente de que os formatos de cadeia customizados não são permitidos em todas as plataformas. Um exemplo disso é que a sequência CipherSpec RC4_SHA_US tem um valor de 05 no IBM i, mas não é uma especificação válida em sistemas AIX, Linux, and Windows. Portanto, se os parâmetros SSLCIPH customizados forem usados em um CLUSRCVR, todos os canais de emissor de cluster autodefinidos resultantes deverão residir nas plataformas nas quais o suporte de TLS subjacente implementa esse CipherSpec e nas quais ele pode ser especificado com o valor customizado. Se não for possível selecionar um valor para o parâmetro SSLCIPH que será entendido em todo o seu cluster, será preciso uma saída de autodefinição de canal para mudá-lo para algo que as plataformas que estão sendo usadas entendam. Use as sequências textuaisCipherSpec quando possível (por exemplo, TLS_RSA_WITH_AES_128_CBC_SHA).

Um parâmetro SSLCRLNL se aplica a um gerenciador de filas individual e não é propagado a outros gerenciadores de filas em um cluster.

Fazendo upgrade de gerenciadores de filas e canais em cluster para SSL/TLS

Faça upgrade dos canais do cluster um por vez, mudando todos os canais CLUSRCVR antes dos canais CLUSSDR.

Antes de começar

Considere as considerações a seguir, como elas podem afetar sua escolha de CipherSpec para um cluster:

- Alguns CipherSpecs não estão disponíveis em todas as plataformas. Tome cuidado ao escolher um CipherSpec que é suportado por todos os gerenciadores de filas no cluster.
- Alguns CipherSpecs podem ser novos na liberação atual do IBM MQ e não são suportados em releases anteriores. Um cluster contendo gerenciadores de filas em execução em diferentes liberações do MQ somente poderá usar os CipherSpecs suportados por cada liberação.

Para usar um novo CipherSpec dentro de um cluster, primeiro deve-se migrar todos os gerenciadores de filas do cluster para a liberação atual.

- Alguns CipherSpecs requerem um tipo específico de certificado digital para ser usado, especialmente aquelas que usam Elliptic Curve Cryptography.

Atenção: Não é possível usar uma combinação de certificados assinados por Curva elíptica e certificados assinados por RSA nos gerenciadores de filas que você deseja associar como parte de um cluster.

Os gerenciadores de filas em um cluster devem todos usar certificados assinados por RSA ou certificados assinados pela EC, não uma mistura de ambos.

Consulte “Certificados digitais e compatibilidade de CipherSpec no IBM MQ” na página 48 para obter mais informações.

Faça upgrade de todos os gerenciadores de filas no cluster para o IBM MQ V8 ou mais recente, se eles já não estiverem nesses níveis. Distribua os certificados e chaves para que o TLS funcione a partir de cada um deles.

Antes de poder fazer upgrade para ou usar qualquer um dos alias CipherSpecs (ANY_TLS13, ANY_TLS13_OR_HIGHER, ANY_TLS12, ANY_TLS12_OR_HIGHERe assim por diante), deve-se fazer upgrade de seus gerenciadores de filas:

- Multi Atualize todos os gerenciadores de filas do IBM MQ for Multiplatforms no cluster para IBM MQ 9.1.4 ou mais recente.
- z/OS Atualize todos os gerenciadores de filas do IBM MQ for z/OS no cluster para IBM MQ for z/OS 9.2.0 ou mais recente.

você deve

Sobre esta tarefa

Mude os canais CLUSRCVR antes dos canais CLUSSDR.

Procedimento

1. Alterne os canais CLUSRCVR para TLS em qualquer ordem desejada, mudando um CLUSRCVR por vez e permita que as mudanças fluam por meio do cluster antes de mudar o próximo.

   Importante: Certifique-se de não alterar o caminho reverso até que as alterações para o canal atual tenham sido distribuídas por todo o cluster.
2. Opcional: Alterne todos os canais CLUSSDR manuais para TLS.

   Isto não possui qualquer efeito na operação do cluster, a menos que você use o comando REFRESH CLUSTER com a opção REPOS(YES).

   Nota: Para clusters grandes, o uso do comando **REFRESH CLUSTER** pode ser disruptivo para o cluster enquanto ele estiver em andamento e novamente em intervalos de 27 dias depois disso, quando os objetos do cluster enviarem automaticamente atualizações de status para todos os gerenciadores de filas interessados Consulte Atualizando em um grande cluster pode afetar o desempenho e disponibilidade do cluster.
3. Use o comando DISPLAY CLUSQMGR para assegurar que a nova configuração de segurança tenha sido propagada por todo o cluster.
4. Reinicie os canais para usar TLS e execute REFRESH SECURITY (SSL).

Conceitos relacionados

“Ativando CipherSpecs” na página 427
Ative um CipherSpec usando o parâmetro **SSLCIPH** no comando do MQSC **DEFINE CHANNEL** ou **ALTER CHANNEL**.

“Certificados digitais e compatibilidade de CipherSpec no IBM MQ” na página 48

Este tópico fornece informações sobre como escolher os certificados digitais e do CipherSpecs para sua política de segurança, delineando o relacionamento entre os certificados digitais e de CipherSpecs no IBM MQ.

Informações relacionadas

Armazenamento em Cluster: Usando Melhores Práticas de REFRESH CLUSTER

Desativando SSL/TLS em gerenciadores de filas e canais em cluster

Para desativar o TLS, configure o parâmetro SSLCIPH como `' '` Desative TLS nos canais de cluster individualmente, mudando todos os canais do receptor de clusters antes dos canais do emissor de clusters.

Sobre esta tarefa

Mude um canal do receptor de clusters por vez e permita que as mudanças fluam através do antes de mudar o próximo.

Importante: Assegure-se de não mudar o caminho reverso até que as mudanças para o canal atual tenham sido distribuídas por todo o cluster.

Procedimento

1. Configure o valor do parâmetro SSLCIPH como `' '`, uma sequência de caracteres vazia em uma aspa única IBM i ou *NONE no IBM i .

   É possível desativar TLS nos canais do receptor de clusters em qualquer ordem desejada.

   Observe que as mudanças fluem na direção oposta nos canais nos quais você deixa o TLS ativo.
2. Verifique se o novo valor é refletido em todos os outros Gerenciadores de Filas usando o comando **DISPLAY CLUSQMGR(*)** ALL.
3. Desligue o TLS em todos os canais do emissor de clusters manuais.

   Isto não possui qualquer efeito na operação do cluster, a menos que você use o comando **REFRESH CLUSTER** com a opção REPOS(YES).

   Para clusters grandes, o uso do comando **REFRESH CLUSTER** pode ser disruptivo para o cluster enquanto ele está em andamento e novamente em intervalos regulares, quando os objetos de cluster enviam automaticamente atualizações de status para todos os gerenciadores de filas interessados. Consulte Atualizando em um cluster grande pode afetar o desempenho e disponibilidade do cluster para obter mais informações.
4. Pare e reinicie os canais do emissor de clusters.

Segurança de Publicação/Assinatura

Os componentes e as interações que estão envolvidos na publicação/assinatura estão descritos como uma introdução para as explicações mais detalhadas e exemplos a seguir.

Existem inúmeros componentes envolvidos na publicação e assinatura para um tópico. Alguns dos relacionamentos de segurança entre eles estão ilustrados em Figura 22 na página 493 e descritos no seguinte exemplo.

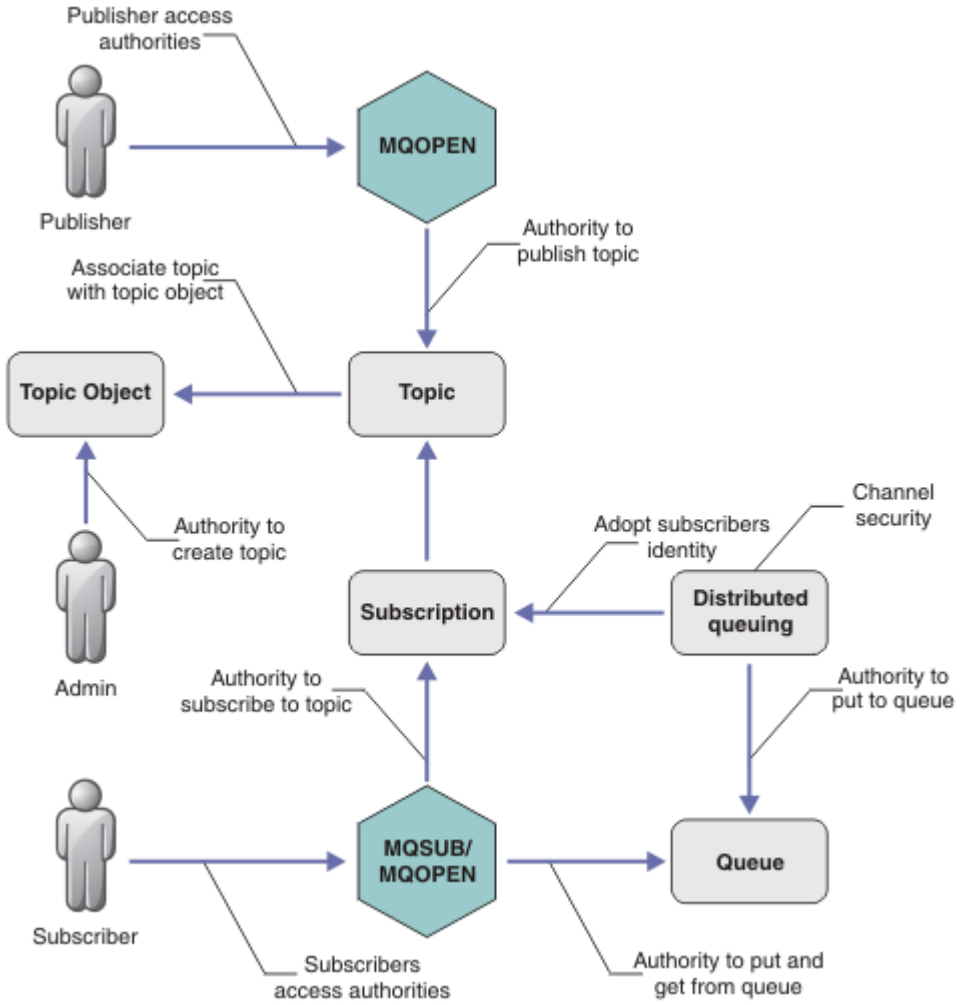


Figura 22. Publicar/assinar relacionamentos de segurança entre vários componentes

Tópicos

Os tópicos são identificados por sequências de tópicos e geralmente organizados em árvores; consulte Árvores de tópicos. É necessário associar um tópico a um objeto do tópico para controlar o acesso ao tópico. O “Modelo de Segurança do Tópico” na página 495 explica como proteger os tópicos usando os objetos do tópico.

Objetos de tópico administrativo

É possível controlar quem tem acesso a um tópico e para qual propósito, usando o comando **setmqaut** com uma lista de objetos de tópico administrativo. Consulte os exemplos, “Conceder acesso a um usuário para assinar um tópico” na página 500 e “Conceder acesso a um usuário para publicar um tópico” na página 507.

z/OS Para controlar o acesso a objetos de tópico no z/OS, consulte Perfis de segurança de tópico

Assinaturas

Subscreva-se em um ou mais tópicos, criando uma assinatura que fornece uma sequência de tópicos, que pode incluir curingas, para corresponder nas sequências de tópicos das publicações. Para obter detalhes adicionais, consulte:

Subscrever usando um objeto do tópico

“Subscrevendo Usando o Nome do Objeto de Tópico” na página 496

Subscrever usando um tópico
"Subscrevendo Usando a Sequência de Tópicos na Qual o Nó de Tópico não Existe" na página 497

Subscrever usando um tópico com curingas
"Subscrevendo Usando uma Sequência de Tópicos que Contém Caracteres Curinga" na página 497

Uma assinatura contém informações sobre a identidade do assinante e a identidade da fila de destino na qual as publicações devem ser colocadas. Ela também contém informações sobre como a publicação deve ser colocada na fila de destino.

Assim como definir quais assinantes possuem autoridade para se subscrever em certos tópicos, é possível restringir as assinaturas para que sejam usadas por um assinante individual. Também é possível controlar quais informações sobre o assinante são usadas pelo gerenciador de filas quando as publicações forem colocadas na fila de destino. Consulte o "Segurança de assinatura" na página 513.

Filas

A fila de destino é uma fila importante para proteger. É local para o assinante e as publicações que correspondiam à assinatura são colocadas nele. Você precisa considerar o acesso à fila de destino a partir de duas perspectivas:

1. Colocando uma publicação na fila de destino.
2. Obtendo a publicação da fila de destino.

O gerenciador de filas coloca uma publicação na fila de destino usando uma identidade fornecida pelo assinante. O assinante ou um programa que delegou a tarefa de obter as publicações, obtém as mensagens da fila. Consulte o "Autoridade para Filas de Destino" na página 498.

Não há nenhum alias de objeto do tópico, mas é possível usar uma fila de alias como o alias para um objeto do tópico. Se fizer isso, e também verificar a autoridade para usar o tópico para publicação ou assinatura, o gerenciador de filas verificará a autoridade para usar a fila.

"Segurança de Publicação/Assinatura entre os Gerenciadores de Filas" na página 514

Sua permissão para publicar ou subscrever em um tópico é verificada no gerenciador de filas locais usando as identidades e autorizações locais. A autorização não depende de o tópico ser definido ou não, nem de onde está definido. Consequentemente, você precisa executar a autorização de tópico em cada gerenciador de filas em um cluster quando os tópicos em cluster forem usados.

Nota: O modelo de segurança para tópicos difere do modelo de segurança para filas. É possível alcançar o mesmo resultado para as filas definindo um alias da fila localmente para cada fila em cluster.

Os gerenciadores de fila trocam assinaturas em um cluster. Na maioria as configurações de cluster do IBM MQ, os canais são configurados com PUTAUT=DEF para colocar mensagens em filas de destino usando a autoridade do processo do canal. É possível modificar a configuração do canal para usar PUTAUT=CTX para requerer que o usuário assinante tenha autoridade para propagar uma assinatura para outro gerenciador de filas em um cluster.

"Segurança de Publicação/Assinatura entre os Gerenciadores de Filas" na página 514 descreve como alterar as suas definições de canal para controlar quem tem permissão para propagar as assinaturas em outros servidores no cluster.

Autorização

É possível aplicar autorização em objetos de tópico, assim como filas e outros objetos. Existem três operações de autorização, `pub`, `sub` e `resume` que permitem a você aplicar apenas nos tópicos. Os detalhes estão descritos em Especificando Autoridades para Diferentes Tipos de Objeto.

Chamadas de função

Nos programas de publicação e assinatura, como em programas enfileirados, as verificações de autorização são feitas quando os objetos são abertos, criados, alterados ou excluídos. As verificações não são feitas quando as chamadas MQPUT ou MQGET MQI são feitas para colocar e obter as publicações.

Para publicar um tópico, execute um MQOPEN no tópico, que executa as verificações de autorização. As mensagens publicadas na manipulação de tópico que usam o comando MQPUT, que não executa nenhuma autorização.

Para subscrever-se em um tópico, geralmente execute um comando MQSUB para criar ou retornar a assinatura e também para abrir a fila de destino para receber as publicações. Como alternativa, execute um MQOPEN separado para abrir a fila de destino e, em seguida, execute o MQSUB para criar ou retomar a assinatura.

Independente das chamadas que você usar, o gerenciador de filas verificar se é possível se subscrever no tópico e obter as publicações resultantes da fila de destino. Se a fila de destino não for gerenciada, as verificações de autorização também serão feitas para que o gerenciador de filas consiga colocar as publicações na fila de destino. Ela usa a identidade que adotou de uma assinatura correspondente. É assumido que o gerenciador de filas sempre consegue colocar as publicações nas filas de destino gerenciadas.

Papéis

Os usuários estão envolvidos em quatro funções na execução de aplicativos de publicação/assinatura:

1. Publicador
2. Assinante
3. Administrador do tópico
4. IBM MQ Administrador - membro do grupo mqm

Defina os grupos com autorizações apropriadas correspondentes às funções de administração do tópico, publicação e assinatura. Em seguida, é possível designar os diretores desses grupos autorizando-os a executar tarefas específicas de publicação e assinatura.

Além disso, você precisa estender as autorizações de operações administrativas para o administrador das filas e canais responsáveis por mover as publicações e assinaturas.

Modelo de Segurança do Tópico

Apenas os objetos de tópico definido possuem atributos de segurança associados. Para uma descrição de objetos de tópico, consulte Objetos de tópico administrativo. Os atributos de segurança especificam se um ID do usuário especificado ou grupo de segurança, tem permissão para executar uma operação de assinatura ou publicação em cada objeto do tópico.

Os atributos de segurança são associados ao nó de administração apropriado na árvore de tópicos. Quando uma verificação de autoridade é feita para um determinado ID do usuário durante uma operação de assinatura ou publicação, a autoridade concedida será baseada nos atributos de segurança do nó associado da árvore de tópicos.

Os atributos de segurança são uma lista de controle de acesso, que indica qual autoridade um determinado ID do usuário do sistema operacional ou grupo de segurança tem para o objeto do tópico.

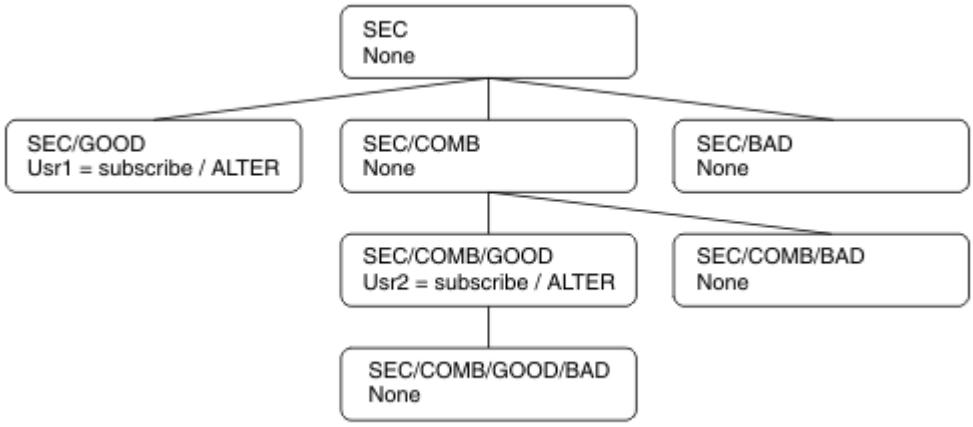
Considere o seguinte exemplo no qual os objetos do tópico foram definidos com os atributos de segurança ou autoridades mostradas:

Tabela 87. Autoridades de Objeto do Tópico de Exemplo

Nome do tópico	**Cadeia do tópico**	**Autoridades-Multiplataformas**	**autoridades do z/OS**
SECROOT	SEC	Nenhum	Nenhum
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD
SECBAD	SEC/BAD	Nenhum	Nenhum HLQ.SUBSCRIBE.SECBAD

Tabela 87. Autoridades de Objeto do Tópico de Exemplo (continuação)			
Nome do tópico	**Cadeia do tópico**	**Autoridades-Multiplataformas**	**autoridades do z/OS**
SECCOMB	SEC/COMB	Nenhum	Nenhum HLQ.SUBSCRIBE.SECCOMB
SECCOMBB	SEC/COMB/GOOD/BAD	Nenhum	Nenhum HLQ.SUBSCRIBE.SECCOMBB
SECCOMBG	SEC/COMB/GOOD	usr2+subscribe	ALTER HLQ.SUBSCRIBE.SECCOMBG
SECCOMBN	SEC/COMB/BAD	Nenhum	Nenhum HLQ.SUBSCRIBE.SECCOMBN

A árvore de tópicos com os atributos de segurança associados em cada nó pode ser representada da seguinte maneira:



Os exemplos listados fornecem as seguintes autorizações:

- No nó-raiz da árvore /SEC, nenhum usuário tem autoridade nesse nó.
- usr1 recebeu autoridade de assinatura para o objeto /SEC/GOOD
- usr2 recebeu autoridade de assinatura para o objeto /SEC/COMB/GOOD

Subscrevendo Usando o Nome do Objeto de Tópico

Ao subscrever em um objeto do tópico especificando o nome MQCHAR48, o nó correspondente na árvore de tópicos está localizado. Se os atributos de segurança associados ao nó indicarem que o usuário tem autoridade para se subscrever, então o acesso será concedido.

Se o usuário não tiver acesso concedido, o nó pai na árvore determinará se o usuário tem autoridade para se subscrever no nível de nó pai. Em caso positivo, o acesso é concedido. Em caso negativo, o pai desse nó será considerado. A recursão continua até que seja localizado um nó que conceda autoridade de assinatura para o usuário. A recursão para quando o nó-raiz é considerado sem que a autoridade tenha sido concedida. No último caso, o acesso é negado.

Em resumo, se algum nó no caminho conceder autoridade para se subscrever nesse usuário ou aplicativo, o assinante terá permissão para se subscrever nesse nó ou em qualquer local abaixo desse nó na árvore de tópicos.

O nó-raiz no exemplo é SEC.

O usuário recebe autoridade de assinatura, se a lista de controle de acesso indicar que o ID do usuário em si tem autoridade ou se um grupo de segurança do sistema operacional do qual o ID do usuário é membro tiver autoridade.

Assim, por exemplo:

- Se `usr1` tentar se subscrever, usando uma sequência de tópicos de `SEC/GOOD`, a assinatura seria alocada porque o ID do usuário tem acesso ao nó associado a esse tópico. No entanto, se `usr1` tentou se subscrever usando a sequência de tópicos `SEC/COMB/GOOD` a assinatura não seria permitida porque o ID do usuário não tem acesso ao nó associado.
- Se o `usr2` tentar se subscrever, usando uma sequência de tópicos de `SEC/COMB/GOOD`, a assinatura seria alocada porque o ID do usuário tem acesso ao nó associado ao tópico. No entanto, se `usr2` tentou se subscrever ao `SEC/GOOD` a assinatura não seria permitida porque o ID do usuário não tem acesso ao nó associado.
- Se `usr2` tentar se subscrever usando uma sequência de tópicos de `SEC/COMB/GOOD/BAD`, a assinatura seria permitida porque o ID do usuário tem acesso ao nó-pai `SEC/COMB/GOOD`.
- Se `usr1` ou `usr2` tentar se subscrever usando uma sequência de tópicos de `/SEC/COMB/BAD`, nenhum seria permitido porque não possuem acesso ao nó de tópico associado, ou nós pais desse tópico.

Uma operação de assinatura que especifica o nome de um objeto do tópico que não existe resulta em um erro de MQRC_UNKNOWN_OBJECT_NAME.

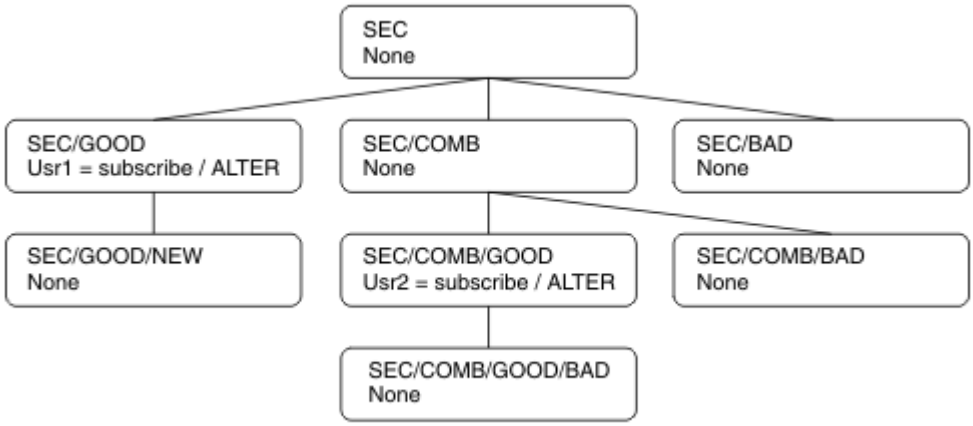
Subscrevendo Usando a Sequência de Tópicos na Qual o Nó de Tópico Existe

O comportamento é igual ao quando especificar o tópico pelo nome do objeto MQCHAR48.

Subscrevendo Usando a Sequência de Tópicos na Qual o Nó de Tópico não Existe

Considere o caso de uma assinatura de aplicativo, especificando uma sequência de tópicos que representa um nó de tópico que não existe atualmente na árvore de tópicos. A verificação de autoridade é executada conforme descrito na seção anterior. A verificação inicia com o nó pai do qual é representado pela sequência de tópicos. Se a autoridade for concedida, um novo nó que representa a sequência de tópicos será criado na árvore de tópicos.

Por exemplo, `usr1` tenta se subscrever em um tópico `SEC/GOOD/NEW`. A autoridade é concedida porque `usr1` tem acesso ao nó-pai `SEC/GOOD`. Um novo nó de tópico é criado na árvore conforme o seguinte diagrama é mostrado. O novo nó de tópico não é um objeto do tópico, ele não tem qualquer atributo de segurança associado diretamente; os atributos são herdados do seu pai.



Subscrevendo Usando uma Sequência de Tópicos que Contém Caracteres Curinga

Considere o caso de subscrever usando uma sequência de tópicos que contém um caractere curinga. A verificação de autoridade é feita no nó na árvore de tópicos que corresponde à parte completa da sequência de tópicos.

Portanto, se um aplicativo se subscrever ao SEC/COMB/GOOD/*, uma verificação de autoridade será executada conforme detalhado nas duas seções anteriores no nó SEC/COMB/GOOD na árvore de tópicos.

De maneira semelhante, se um aplicativo precisar ser subscrever ao SEC/COMB/*/GOOD, uma verificação de autoridade será executada no nó SEC/COMB.

Autoridade para Filas de Destino

Ao se inscrever em um tópico, um dos parâmetros é o identificador hobj de uma fila que foi aberta para saída para receber as publicações.

Se hobj não for especificado, mas estiver em branco, uma fila gerenciada será criada se as condições a seguir se aplicarem:

- A opção MQSO_MANAGED foi especificada.
- A assinatura não existe.
- A criação é especificada.

Se hobj estiver em branco e você estiver mudando ou retomando uma assinatura existente, a fila de destino fornecida anteriormente poderá ser gerenciada ou não gerenciada.

O aplicativo ou o usuário que faz a solicitação MQSUB deve ter a autoridade para colocar as mensagens na fila de destino que forneceu; na realidade, a autoridade para fazer com que as mensagens publicadas sejam colocadas nessa fila. A verificação de autoridade segue as regras existentes para a verificação de segurança da fila.

A verificação de segurança inclui verificações alternativas de contexto e ID do usuário onde necessário. Para conseguir configurar alguns dos campos de contexto de Identidade você deve especificar a opção MQSO_SET_IDENTITY_CONTEXT bem como a opção MQSO_CREATE ou MQSO_ALTER. Não é possível configurar qualquer um dos campos de contexto de Identidade em uma solicitação MQSO_RESUME.

Se o destino for uma fila gerenciada, nenhuma verificação de segurança será executada no destino gerenciado. Se tiver permissão para se subscrever em um tópico, é assumido você pode usar os destinos gerenciados.

Publicando Usando o Nome do Tópico ou Sequência de Tópicos Onde Existir o Nó de Tópico

O modelo de segurança para publicação é o mesmo que para assinatura, com exceção dos curingas. Publicações não contêm curingas; portanto, não há nenhum caso de uma sequência de tópicos contendo curingas para ser considerado.

As autoridades para publicar e subscrever são distintas. Um usuário ou grupo pode ter a autoridade para executar uma sem conseguir necessariamente executar a outra.

Ao publicar em um objeto do tópico especificando o nome MQCHAR48 ou a sequência de tópicos, o nó correspondente na árvore de tópicos será localizado. Se os atributos de segurança associados ao nó de tópico indicarem que o usuário tem autoridade para publicar, o acesso será concedido.

Se o acesso não for concedido, o nó pai na árvore determinará se o usuário tem autoridade para publicar nesse nível. Em caso positivo, o acesso é concedido. Em caso negativo, a recursão continuará até que seja localizado um nó que conceda autoridade de publicação ao usuário. A recursão para quando o nó-raiz é considerado sem que a autoridade tenha sido concedida. No último caso, o acesso é negado.

Em resumo, se algum nó no caminho conceder autoridade para publicar nesse usuário ou aplicativo, o publicador terá permissão para publicar nesse nó ou em qualquer lugar abaixo desse nó na árvore de tópicos.

Publicando Usando o Nome do Tópico ou Sequência de Tópicos Onde não Existir o Nó de Tópico

Assim como na operação de assinatura, quando um aplicativo é publicado, especificando uma sequência de tópicos que representa um nó de tópico que não existe atualmente na árvore de tópicos, a verificação de autoridade é executada iniciando com o pai do nó representado pela sequência de tópicos. Se a autoridade for concedida, um novo nó que representa a sequência de tópicos será criado na árvore de tópicos.

Publicando Usando uma Fila de Alias que Resolve em um Objeto de Tópico

Se você publicar usando uma fila de alias que é resolvida em um objeto de tópico, a verificação de segurança ocorrerá na fila de alias e no tópico subjacente no qual é resolvido.

A verificação de segurança na fila de alias verifica se o usuário tem autoridade para colocar as mensagens nessa fila de alias e a verificação de segurança no tópico verifica se o usuário pode publicar nesse tópico. Quando uma fila de alias é resolvida para outra fila, as verificações não são feitas na fila subjacente. A verificação de autoridade é executada de maneira diferente para os tópicos e as filas.

Fechando uma Assinatura

Ocorre uma verificação de segurança adicional se você fechar a assinatura usando a opção `MQCO_REMOVE_SUB` se não criar a assinatura sob esta manipulação.

Uma verificação de segurança é executada para assegurar que tenha a autoridade correta para fazer isso porque a ação resulta na remoção da assinatura. Os atributos de segurança associados ao nó de tópico indicam que o usuário tem autoridade e, em seguida, o acesso é concedido. Em caso negativo, o nó-pai na árvore é considerado para determinar se o usuário tem autoridade para fechar a assinatura. A recursão continua até que a autoridade seja concedida ou o nó-raiz seja atingido.

Definindo, Alterando e Excluindo uma Assinatura

Nenhuma verificação de segurança de assinatura é executada quando uma assinatura for criada administrativamente, em vez de usar uma solicitação de API `MQSUB`. O administrador já recebeu esta autoridade por meio do comando.

As verificações de segurança são executadas para assegurar que as publicações podem ser colocadas na fila de destino associadas à assinatura. As verificações são executadas da mesma maneira que para uma solicitação `MQSUB`.

O ID do usuário que é usado para essas verificações de segurança depende do comando sendo emitido. Se o parâmetro **SUBUSER** for especificado, ele afetará a maneira como a verificação é executada, conforme mostrado em Tabela 88 na página 499:

Tabela 88. IDs de Usuário Usados para Verificações de Segurança para Comandos

Comando:	SUBUSER especificado e em branco	SUBUSER especificado e concluído	SUBUSER não especificado
	Use o ID de administrador		Use o ID do usuário a partir da assinatura LIKE

Tabela 88. IDs de Usuário Usados para Verificações de Segurança para Comandos (continuação)			
Comando:	**SUBUSER especificado e em branco**	**SUBUSER especificado e concluído**	**SUBUSER não especificado**
	Use o ID de administrador		Use o ID do.DEFAULT.SU usuário aB - se estiver partir daem branco, assinaturause o ID de SYSTEMadministrador
	Use o ID de administrador		Use o ID do usuário a partir da assinatura existente

A única verificação de segurança executada ao excluir as assinaturas usando o comando DELETE SUB é a verificação de segurança de comando.

Exemplo de configuração de segurança de publicação/assinatura

Esta seção descreve um cenário que tem o controle de acesso configurado em tópicos de uma maneira que permita que o controle de segurança seja aplicado conforme necessário.

Conceder acesso a um usuário para assinar um tópico

Este tópico é o primeiro em uma lista de tarefas que informam como conceder acesso aos tópicos por mais de um usuário.

Sobre esta tarefa

Esta tarefa assume que nenhum objeto de tópico administrativo existe nem quaisquer perfis foram definidos para assinatura ou publicação. Os aplicativos estão criando novas assinaturas em vez de continuar as existentes e estão fazendo isso usando somente a sequência de tópicos.

Um aplicativo pode fazer uma assinatura fornecendo um objeto do tópico, uma sequência de tópicos ou uma combinação de ambos. Independentemente do caminho o aplicativo selecione, o efeito é fazer uma assinatura em um determinado ponto na árvore de tópicos. Se este ponto na árvore de tópicos for representado por um objeto de tópico administrativo, um perfil de segurança será verificado com base no nome do objeto do tópico.

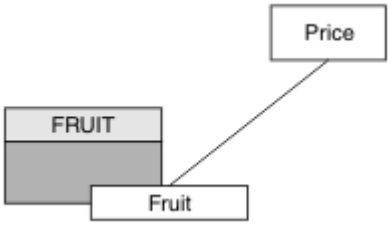


Figura 23. Exemplo de acesso do objeto do tópico

Tabela 89. Acesso ao objeto do tópico de exemplo		
Tópico	**Acesso de assinatura necessário**	**Objeto do Tópico**
Price	Nenhum usuário	Nenhum
Preço/Fruta	USER1	FRUTA

Defina um novo objeto do tópico conforme a seguir:

Procedimento

1. Emita o comando MQSC `DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit')`.
2. Conceda acesso como a seguir:

- z/OS **z/OS** :

  Conceda acesso a USER1 para assinar o tópico "`Price/Fruit`" concedendo o acesso do usuário ao perfil `hlq.SUBSCRIBE.FRUIT`. Faça isso usando os comandos RACF a seguir:

```
RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
```

- Multi **Multiplataformas**:

  Conceda acesso a USER1 para assinar o tópico "`Price/Fruit`" concedendo o acesso do usuário ao objeto FRUIT. Faça isso usando o comando de autorização para a plataforma:

  ALW **Sistemas AIX, Linux, and Windows**

```
setmqaut -t topic -n FRUIT -p USER1 +sub
```

  IBM i **IBM i**

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
```

Resultados

Quando USER1 tenta assinar o tópico "`Price/Fruit`" o resultado é sucesso.

Quando USER2 tenta assinar o tópico "`Price/Fruit`" o resultado é uma falha com uma mensagem `MQRC_NOT_AUTHORIZED`, juntamente com:

- z/OS No z/OS, as mensagens a seguir vistas no console que mostram o caminho de segurança completo através da árvore de tópicos que foi tentada:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW No AIX, Linux, and Windows, o evento de autorização a seguir:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
```

```
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i No IBMi, o evento de autorização a seguir:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

Observe que esta é uma ilustração do que você vê; não todos os campos.

Conceder acesso a um usuário para assinar um tópico mais fundo na árvore

Este tópico é o segundo em uma lista de tarefas que informam como conceder acesso aos tópicos por mais de um usuário.

Antes de começar

Este tópico usa a configuração descrita em "Conceder acesso a um usuário para assinar um tópico" na página 500.

Sobre esta tarefa

Se o ponto na árvore de tópicos no qual o aplicativo faz a assinatura não é representado por um objeto de tópico administrativo, mova para cima na árvore até que o objeto do tópico administrativo pai mais próximo seja localizado. O perfil de segurança é verificado com base no nome do objeto do tópico.

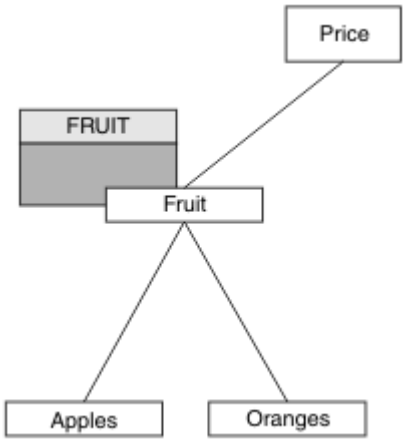


Figura 24. Exemplo de concessão de acesso a um tópico em uma árvore de tópicos

Tabela 90. Requisitos de acesso para tópicos de exemplo e objetos de tópico

Tópico	Acesso de assinatura necessário	Objeto do Tópico
Price	Nenhum usuário	Nenhum
Preço/Fruta	USER1	FRUTA
Preço/Fruta/Maçãs	USER1	
Preço/Fruta/Laranjas	USER1	

No “Conceder acesso a um usuário para assinar um tópico” na página 500, USER1 foi concedido acesso para assinar o tópico `"Price/Fruit"` concedendo a ele acesso ao perfil `hlq.SUBSCRIBE.FRUIT` no z/OS e assinar acesso ao perfil FRUIT em Multiplataformas. Esse perfil único também concede acesso para assinar USER1 para `"Price/Fruit/Apples"`, `"Price/Fruit/Oranges"` e `"Price/Fruit/#"`.

Quando USER1 tenta assinar o tópico `"Price/Fruit/Apples"` o resultado é sucesso.

Quando USER2 tenta assinar o tópico `"Price/Fruit/Apples"` o resultado é uma falha com uma mensagem `MQRC_NOT_AUTHORIZED`, juntamente com:

- z/OS No z/OS, as mensagens a seguir vistas no console que mostram o caminho de segurança completo através da árvore de tópicos que foi tentada:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- Multi Em Multiplataformas, o seguinte evento de autorização:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

Observe o seguinte :

- z/OS As mensagens que você recebe no z/OS são idênticas àquelas recebidas na tarefa anterior como os mesmos objetos e perfis de objeto estão controlando o acesso.
- Multi A mensagem de evento recebida em Multiplataformas é semelhante à recebida na tarefa anterior, mas a sequência de tópicos real é diferente.

Garanta acesso a outro usuário para assinar somente o tópico dentro da árvore

Este tópico é o terceiro em uma lista de tarefas que informam como conceder acesso para assinar tópicos por mais de um usuário.

Antes de começar

Este tópico usa a configuração descrita em “Conceder acesso a um usuário para assinar um tópico mais fundo na árvore” na página 502.

Sobre esta tarefa

No “Conceder acesso a um usuário para assinar um tópico mais fundo na árvore” na página 502, USER2 foi recusado o acesso ao tópico `"Price/Fruit/Apples"`. Esse tópico informa como conceder acesso a esse tópico, mas não para quaisquer outros tópicos.

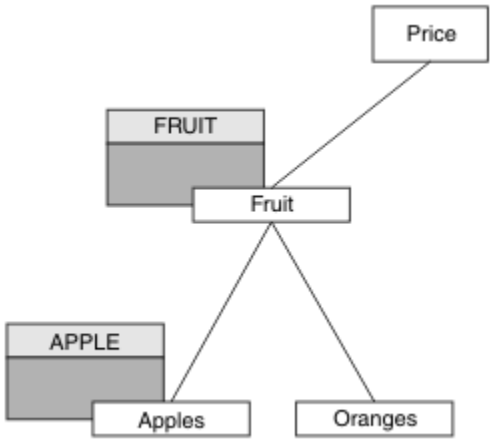


Figura 25. Concedendo acesso a tópicos específicos em uma árvore de tópicos

Tabela 91. Requisitos de acesso para tópicos de exemplo e objetos de tópico

Tópico	Acesso de assinatura necessário	Objeto do Tópico
Price	Nenhum usuário	Nenhum
Preço/Fruta	USER1	FRUTA
Preço/Fruta/Maçãs	USER1 e USER2	APPLE
Preço/Fruta/Laranjas	USER1	

Defina um novo objeto do tópico conforme a seguir:

Procedimento

1. Emita o comando MQSC `DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')`.
2. Conceda acesso como a seguir:

- z/OS **z/OS** :

  No “Conceder acesso a um usuário para assinar um tópico mais fundo na árvore” na página 502 USER1 foi concedido acesso para assinar o tópico `"Price/Fruit/Apples"` concedendo ao usuário acesso ao perfil `hlq.SUBSCRIBE.FRUIT` .

  Esse único perfil também concedeu acesso USER1 para assinar o `"Price/Fruit/Oranges"` `"Price/Fruit/#"` e esse acesso permanece mesmo com a inclusão do novo objeto do tópico e os perfis associados.

  Conceder acesso a USER2 para assinar o tópico `"Price/Fruit/Apples"` concedendo o acesso do usuário ao perfil `hlq.SUBSCRIBE.APPLE`. Faça isso usando os comandos RACF a seguir:

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
  PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
  ```

- Multi Multiplataformas:

  No “Conceder acesso a um usuário para assinar um tópico mais fundo na árvore” na página 502 USER1 foi concedido acesso para assinar o tópico `"Price/Fruit/Apples"` concedendo ao usuário acesso de assinatura para o perfil `FRUIT` .

Este perfil único também concedeu acesso USER1 para acessar "Price/Fruit/Oranges" e "Price/Fruit/#" e esse acesso permanece mesmo com a inclusão do novo objeto do tópico e os perfis associados a ele.

Conceda acesso a USER2 para assinar o tópico "Price/Fruit/Apples" concedendo o acesso à assinatura do usuário para o perfil APPLE. Faça isso usando o comando de autorização para a plataforma:

ALW **Sistemas AIX, Linux, and Windows**

```
setmqaut -t topic -n APPLE -p USER2 +sub
```

IBM i **IBM i**

```
GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*SUB)
```

Resultados

z/OS No z/OS, quando USER1 tenta assinar o tópico "Price/Fruit/Apples" a primeira verificação de segurança no perfil hlq.SUBSCRIBE.APPLE falha, mas ao mover a árvore do perfil para cima o perfil hlq.SUBSCRIBE.FRUIT permite que o USER1 assine, portanto, a assinatura é bem-sucedida e nenhum código de retorno será enviado para a chamada MQSUB. No entanto, uma mensagem RACF ICH é gerada para a primeira verificação:

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

Quando USER2 tenta assinar o tópico "Price/Fruit/Apples" o resultado é sucesso porque a verificação de segurança aprova o primeiro perfil.

Quando USER2 tenta assinar o tópico "Price/Fruit/Oranges" o resultado será uma falha com uma mensagem MQRC_NOT_AUTHORIZED juntamente com:

- z/OS No z/OS, as mensagens a seguir vistas no console que mostram o caminho de segurança completo através da árvore de tópicos que foi tentada:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW Nas plataformas AIX, Linux, and Windows, o evento de autorização a seguir:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

- IBM i No IBMi, o evento de autorização a seguir:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

z/OS A desvantagem dessa configuração é que, no z/OS, você receberá mensagens ICH adicionais no console. É possível evitar isso se você proteger a árvore de tópicos de maneira diferente.

Mudar o controle de acesso para evitar mensagens adicionais

Este tópico é o quarto em uma lista de tarefas que informa como conceder acesso para assinar tópicos por mais de um usuário e evitar mensagens adicionais RACF ICH408I no z/OS.

Antes de começar

Este tópico aprimora a configuração descrita em "Garanta acesso a outro usuário para assinar somente o tópico dentro da árvore" na página 503 para que você evite mensagens de erro adicionais.

Sobre esta tarefa

Esse tópico informa como conceder acesso a tópicos mais profundos na árvore e como remover o acesso ao tópico para baixo na árvore quando nenhum usuário requerer isso.

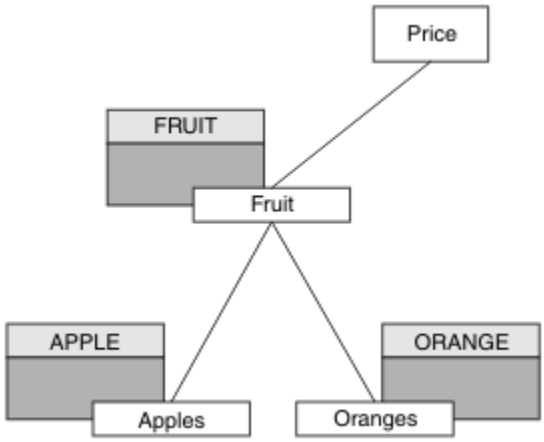


Figura 26. Exemplo de concessão do controle de acesso para evitar mensagens adicionais.

Defina um novo objeto do tópico conforme a seguir:

Procedimento

1. Emita o comando MQSC `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')`.
2. Conceda acesso como a seguir:

   - z/OS **z/OS** :

     Defina um novo perfil e inclua acesso àquele perfil e aos perfis existentes. Faça isso usando os comandos RACF a seguir:

     ```
     RDEFINE MXTOPIC hlq.SUBSCRIBE.ORANGE UACC(NONE)
     PERMIT hlq.SUBSCRIBE.ORANGE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
     PERMIT hlq.SUBSCRIBE.APPLE CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
     ```

   - Multi Multiplataformas:

     Configure o acesso equivalente usando os comandos de autorização para a plataforma:

     ALW **Sistemas AIX, Linux, and Windows**

     ```
     setmqaut -t topic -n ORANGE -p USER1 +sub
     setmqaut -t topic -n APPLE -p USER1 +sub
     ```

IBM i

```
GRTMQAUT OBJ(ORANGE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
```

Resultados

z/OS No z/OS, quando USER1 tenta assinar o tópico `"Price/Fruit/Apples"` a primeira verificação de segurança no perfil `hlq.SUBSCRIBE.APPLE` é bem-sucedida.

Quando USER2 tenta assinar o tópico `"Price/Fruit/Apples"` o resultado é sucesso porque a verificação de segurança aprova o primeiro perfil.

Quando USER2 tenta assinar o tópico `"Price/Fruit/Oranges"` o resultado será uma falha com uma mensagem `MQRC_NOT_AUTHORIZED` juntamente com:

- z/OS No z/OS, as mensagens a seguir vistas no console que mostram o caminho de segurança completo através da árvore de tópicos que foi tentada:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.ORANGE ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW No AIX, Linux, and Windows, o evento de autorização a seguir:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- IBM i No IBM i, o evento de autorização a seguir:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

Conceder acesso a um usuário para publicar um tópico

Este tópico é o primeiro uma em uma lista de tarefas que informam como conceder acesso para publicar os tópicos por mais de um usuário.

Sobre esta tarefa

Esta tarefa assume que nenhum objeto de tópico administrativo exista no lado direito da árvore de tópicos, nem quaisquer perfis foram definidos para publicação. A suposição usada é que os publicadores estão usando somente a sequência de tópicos.

Um aplicativo pode publicar em um tópico fornecendo um objeto do tópico, uma sequência de tópicos ou uma combinação de ambos. Qualquer forma o aplicativo seleciona, o efeito é publicar em um determinado ponto na árvore de tópicos. Se este ponto na árvore de tópicos for representado por um objeto de tópico administrativo, um perfil de segurança será verificado com base no nome do objeto do tópico. Por exemplo:

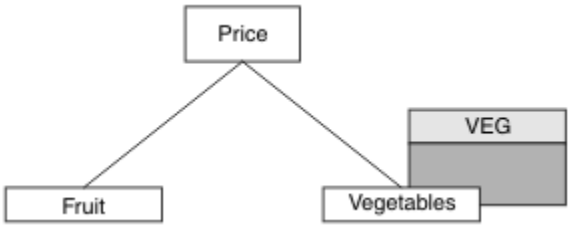


Figura 27. Concedendo acesso de publicação a um tópico

Tabela 92. Exemplo de requisitos de acesso de publicação

Tópico	Acesso de publicação necessário	Objeto do Tópico
Price	Nenhum usuário	Nenhum
Preço/Vegetais	USER1	VEG

Defina um novo objeto do tópico conforme a seguir:

Procedimento

1. Emita o comando MQSC `DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')`.
2. Conceda acesso como a seguir:

- z/OS **z/OS** :

  Conceda acesso para USER1 para publicar no tópico `"Price/Vegetables"` concedendo o acesso do usuário ao perfil `hlq.PUBLISH.VEG`. Faça isso usando os comandos RACF a seguir:

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
  PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
  ```

- Em outras plataformas:

  Conceda acesso para USER1 para publicar no tópico `"Price/Vegetables"` concedendo o acesso do usuário ao perfil VEG. Faça isso usando o comando de autorização para a plataforma:

  ALW **Sistemas AIX, Linux, and Windows**

  ```
  setmqaut -t topic -n VEG -p USER1 +pub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
  ```

Resultados

Quando USER1 tenta publicar no tópico `"Price/Vegetables"` o resultado é sucesso; ou seja, a chamada MQOPEN é bem-sucedida.

Quando USER2 tentar publicar no tópico `"Price/Vegetables"` a chamada MQOPEN falhará com uma mensagem `MQRC_NOT_AUTHORIZED`, juntamente com:

- z/OS No z/OS, as mensagens a seguir vistas no console que mostram o caminho de segurança completo através da árvore de tópicos que foi tentada:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ALW Em outras plataformas, o evento de autorização a seguir:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

- IBM i No IBMi, o evento de autorização a seguir:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

Observe que esta é uma ilustração do que você vê; não todos os campos.

Conceder acesso a um usuário para publicar em um tópico dentro da árvore

Este tópico é o segundo em uma lista de tarefas que informam como conceder acesso a publicação em tópicos por mais de um usuário.

Antes de começar

Este tópico usa a configuração descrita em “Conceder acesso a um usuário para publicar um tópico” na página 507.

Sobre esta tarefa

Se o ponto na árvore de tópicos no qual o aplicativo publicar não for representado por um objeto do tópico administrativo, mova a árvore para cima até onde o objeto do tópico administrativo pai mais próximo está localizado. O perfil de segurança é verificado com base no nome do objeto do tópico.

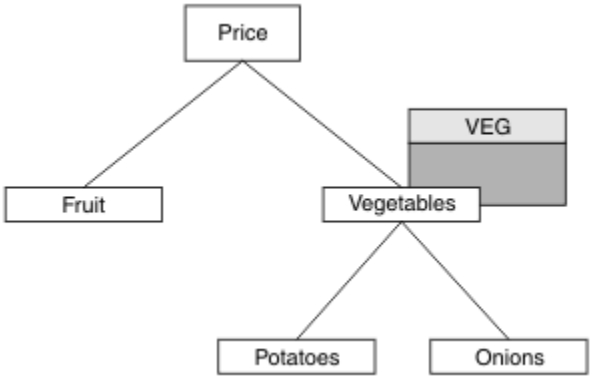


Figura 28. Concedendo acesso de publicação em um tópico em uma árvore de tópicos

Tabela 93. Exemplo de requisitos de acesso de publicação		
Tópico	**Acesso de assinatura necessário**	**Objeto do Tópico**
Price	Nenhum usuário	Nenhum
Preço/Vegetais	USER1	VEG
Preço/Vegetais/Batatas	USER1	
Preço/Vegetais/Cebolas	USER1	

Na tarefa anterior, USER1 recebeu acesso para publicar tópico `"Price/Vegetables/Potatoes"` concedendo a ele acesso ao perfil `hlq.PUBLISH.VEG` no z/OS ou acesso de publicação ao VEG perfil em Multiplataformas. Esse único perfil também concede acesso USER1 para publicação em `"Price/Vegetables/Onions"`.

Quando USER1 tenta publicar no tópico `"Price/Vegetables/Potatoes"` o resultado é sucesso; essa chamada MQOPEN será bem-sucedida.

Quando USER2 tenta assinar o tópico `"Price/Vegetables/Potatoes"` o resultado é uma falha; ou seja, a chamada MQOPEN falhará com uma mensagem `MQRC_NOT_AUTHORIZED`, juntamente com:

- z/OS No z/OS, as mensagens a seguir vistas no console que mostram o caminho de segurança completo através da árvore de tópicos que foi tentada:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Multi Em Multiplataformas, o seguinte evento de autorização:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

Observe o seguinte :

- z/OS As mensagens que você recebe no z/OS são idênticas àquelas recebidas na tarefa anterior como os mesmos objetos e perfis de objeto estão controlando o acesso.
- Multi A mensagem de evento recebida em Multiplataformas é semelhante à recebida na tarefa anterior, mas a sequência de tópicos real é diferente.

Conceder acesso para publicação e assinatura

Este tópico é o último em uma lista de tarefas que informam como conceder acesso para publicar e assinar tópicos por mais de um usuário.

Antes de começar

Este tópico usa a configuração descrita em “Conceder acesso a um usuário para publicar em um tópico dentro da árvore” na página 509.

Sobre esta tarefa

Em uma tarefa anterior USER1 foi fornecido acesso para assinar o tópico `"Price/Fruit"`. Este tópico informa como conceder acesso ao usuário para publicar para o tópico.

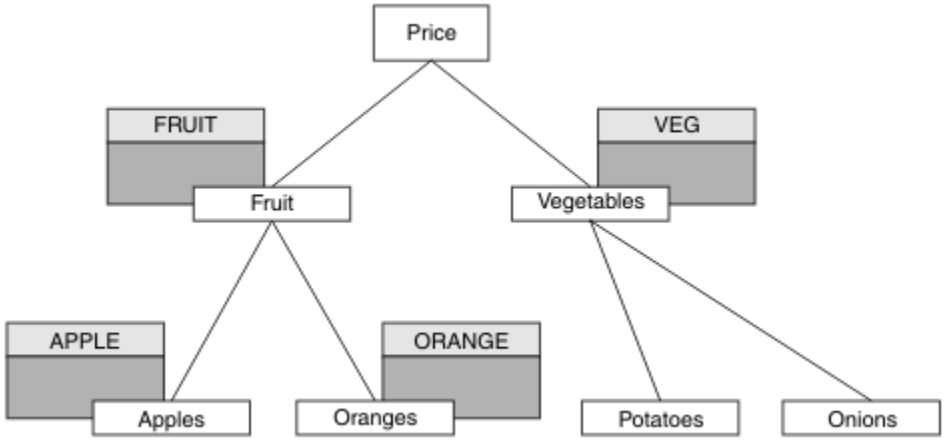


Figura 29. Concedendo acesso para publicação e assinatura

Tabela 94. Exemplo de requisitos de acesso de publicação e assinatura

Tópico	Acesso de assinatura necessário	Acesso de publicação necessário	Objeto do Tópico
Price	Nenhum usuário	Nenhum usuário	Nenhum
Preço/Fruta	USER1	USER1	FRUTA
Preço/Fruta/Maçãs	USER1 e USER2		APPLE
Preço/Fruta/Laranjas	USER1		LARANJA

Procedimento

Conceda acesso como a seguir:

- z/OS **z/OS** :

  Em uma tarefa anterior USER1 recebeu acesso para assinar o tópico `"Price/Fruit"`, concedendo o acesso do usuário ao perfil `hlq.SUBSCRIBE.FRUIT`.

  Para publicar o tópico `"Price/Fruit"` conceda acesso ao USER1 para o perfil `hlq.PUBLISH.FRUIT`. Faça isso usando os comandos RACF a seguir:

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
  PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Multi Multiplataformas:

  Conceda acesso para USER1 para publicar no tópico `"Price/Fruit"` por conceder o acesso de publicação do usuário para o perfil FRUIT. Faça isso usando o comando de autorização para a plataforma:

ALW Sistemas AIX, Linux, and Windows

```
setmqaut -t topic -n FRUIT -p USER1 +pub
```

IBM i IBM i

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
```

Resultados

z/OS No z/OS, quando USER1 tentar de publicar no tópico `"Price/Fruit"` a verificação de segurança na chamada MQOPEN aprova.

Quando USER2 tentar publicar no tópico `"Price/Fruit"`, o resultado será uma falha com uma mensagem `MQRC_NOT_AUTHORIZED`, juntamente com:

- z/OS No z/OS, as mensagens a seguir vistas no console que mostram o caminho de segurança completo através da árvore de tópicos que foi tentada:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ALW Nas plataformas AIX, Linux, and Windows, o evento de autorização a seguir:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i No IBM i, o evento de autorização a seguir:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

Seguindo o conjunto completo dessas tarefas, fornece USER1 e USER2 as seguintes autoridades de acesso para publicar e subscrever-se nos tópicos listados:

Tabela 95. Conclua lista de autoridades de acesso resultante de exemplos de segurança

Tópico	Acesso de assinatura necessário	Acesso de publicação necessário	Objeto do Tópico
Price	Nenhum usuário	Nenhum usuário	Nenhum
Preço/Fruta	USER1	USER1	FRUTA
Preço/Fruta/Maçãs	USER1 e USER2		APPLE
Preço/Fruta/Laranjas	USER1		LARANJA

Tabela 95. Conclua lista de autoridades de acesso resultante de exemplos de segurança (continuação)			
Tópico	**Acesso de assinatura necessário**	**Acesso de publicação necessário**	**Objeto do Tópico**
Preço/ Vegetais		USER1	VEG
Preço/ Vegetais/ Batatas			
Preço/ Vegetais/ Cebolas			

z/OS Onde houver requisitos diferentes para acesso de segurança em níveis diferentes dentro da árvore de tópicos, o planejamento cuidadoso assegura que você não receberá avisos de segurança externos no log do console do z/OS. Configurando a segurança no nível correto na árvore evita mensagens de segurança enganosas.

Segurança de assinatura

MQSO_ALTERNATE_USER_AUTHORITY

O campo AlternateUserId contém um identificador de usuário a ser usado para validar esta chamada MQSUB. A chamada pode ser bem-sucedida somente se este AlternateUserId estiver autorizado a assinar o tópico com opções de acesso especificadas, independentemente de se o ID do usuário sob o qual o aplicativo estiver em execução está autorizado para tanto.

MQSO_SET_IDENTITY_CONTEXT

A assinatura deve usar o token de conta e os dados de identificação do aplicativo fornecidos nos campos PubAccountingToken e PubApplIdentityData.

Se esta opção for especificada, a mesma verificação de autorização será executada como se a fila de destino tivesse sido acessada usando uma chamada MQOPEN com MQOO_SET_IDENTITY_CONTEXT, exceto no caso em que a opção MQSO_MANAGED também é usada; neste caso, não há nenhuma verificação de autorização na fila de destino.

Se esta opção não for especificada, as publicações enviadas a este assinante terão informações de contexto padrão associadas a elas da seguinte maneira:

Tabela 96. Informações de contexto de publicação padrão	
Campo no MQMD	**Valor Usado**
UserIdentifier	O ID do usuário associado à assinatura (consulte o campo SUBUSER em DISPLAY SBSTATUS) no momento em que a publicação é feita.
AccountingToken	Determinado a partir do ambiente, se possível; caso contrário, configure como MQACT_NONE.
ApplIdentityData	Configure como em branco.

Esta opção é válida apenas com MQSO_CREATE e MQSO_ALTER. Se usada com MQSO_RESUME, os campos PubAccountingToken e PubApplIdentityData são ignorados, portanto, esta opção não tem efeito.

Se uma assinatura é alterada sem o uso dessa opção na qual a opção forneceu informações de contexto anteriormente, as informações de contexto padrão são geradas para assinatura alterada.

Se uma assinatura permitindo que diferentes IDs de usuário a usem com a opção MQSO_ANY_USERID for continuada por um ID do usuário diferente, o contexto de identidade padrão será gerado para o novo ID do usuário que agora possui a assinatura e todas as publicações subsequentes serão entregues contendo o novo contexto de identidade.

AlternateSecurityId

Este é um identificador de segurança que é transmitido com o AlternateUserId para o serviço de autorização para permitir que verificações de autorização apropriadas sejam executadas. AlternateSecurityId é usado somente se MQSO_ALTERNATE_USER_AUTHORITY for especificado e o campo AlternateUserId não estiver completamente em branco até o primeiro caractere nulo ou no final do campo.

Opção de Assinatura MQSO_ANY_USERID

Quando MQSO_ANY_USERID é especificado, a identidade do assinante não é restrita a um único ID do usuário. Isso permite que qualquer usuário altere ou continue a assinatura quando tem autoridade adequada. Apenas um único usuário pode ter a assinatura a qualquer momento. Uma tentativa de continuar o uso de uma assinatura atualmente em uso por outro aplicativo irá fazer com que a chamada falhe com MQRC_SUBSCRIPTION_IN_USE.

Para incluir essa opção a uma assinatura existente a chamada MQSUB (utilizando MQSO_ALTER) deve vir do mesmo ID do usuário da assinatura original.

Se uma chamada MQSUB se referir a uma assinatura existente com MQSO_ANY_USERID configurado e o ID do usuário for diferente da assinatura original, a chamada será bem-sucedida apenas se o novo ID do usuário tiver autoridade para assinar o tópico. Após concluir com sucesso, publicações futuras a este assinante serão colocadas na fila do assinante com o novo ID do usuário configurado na publicação.

MQSO_FIXED_USERID

Quando MQSO_FIXED_USERID é especificado, a assinatura somente pode ser alterada ou retomada por um único ID do usuário proprietário. Este ID do usuário é o último ID do usuário a alterar a assinatura que configura esta opção, removendo a opção MQSO_ANY_USERID ou se nenhuma alteração ocorreu, é o ID do usuário que criou a assinatura.

Se um verbo MQSUB se referir a uma assinatura existente com MQSO_ANY_USERID configurado e altera a assinatura (usando MQSO_ALTER) para usar a opção MQSO_FIXED_USERID, o ID do usuário da assinatura agora será fixo nesse novo ID do usuário. A chamada será bem-sucedida apenas se o novo ID do usuário tiver autoridade para assinar o tópico.

Se um ID do usuário diferente do registrado como pertencente a uma assinatura tenta continuar ou alterar uma assinatura MQSO_FIXED_USERID, a chamada falhará com MQRC_IDENTITY_MISMATCH. O ID do usuário proprietário de uma assinatura pode ser visualizado usando o comando DISPLAY SBSTATUS.

Se nem MQSO_ANY_USERID ou MQSO_FIXED_USERID forem especificados, o padrão será MQSO_FIXED_USERID.

Segurança de Publicação/Assinatura entre os Gerenciadores de Filas

Mensagens internas de publicação/assinatura, como assinaturas de proxy e publicações são colocadas em filas do sistema de publicação/assinatura usando as regras de segurança de canal normal. As informações e diagramas neste tópico destacam os vários processos e IDs de usuário envolvidos na entrega dessas mensagens.

Controle de Acesso Local

O acesso aos tópicos para publicação e assinaturas é controlado pelas definições de segurança local e regras que são descritas em Segurança de Publicação/Assinatura. Nenhum objeto de tópico local é necessário para estabelecer controle de acesso. Os administradores podem optar por aplicar o controle de acesso aos objetos do tópico em cluster, independentemente se eles existirem no cluster ainda.

Os administradores do sistema são responsáveis pelo controle de acesso em seu sistema local. Eles devem confiar nos administradores de outros membros da hierarquia ou dos coletivos do cluster para serem responsáveis pela política de controle de acesso. Como o controle de acesso é definido para cada máquina separada, é provável que seja oneroso se o controle de nível bom for necessário. Pode não ser necessário impor qualquer controle de acesso ou o controle de acesso pode ser definido nos objetos de alto nível na árvore de tópicos. O controle de acesso de nível de multa pode ser definido para cada subdivisão do espaço de nomes de tópico.

Fazendo uma Assinatura de Proxy

Confie em uma organização para conectar seu gerenciador de filas em seu gerenciador de filas é confirmado por meio de autenticação de canal normal. Se essa organização de confiança também for permitida para fazer publicação/assinatura distribuída, uma verificação de autoridade será feita. A verificação é feita quando o canal coloca uma mensagem em uma fila de publicação/assinatura distribuída. Por exemplo, se uma mensagem for colocada para a fila `SYSTEM.INTER.QMGR.CONTROL`. O ID do usuário para a verificação de autoridade da fila depende dos valores PUTAUT do canal de recebimento. Por exemplo, o ID do usuário do canal, `MCAUSER`, o contexto da mensagem, dependendo do valor e da plataforma. Para obter mais informações sobre a segurança do canal, consulte Segurança de Canal.

As assinaturas de proxy são feitas com o ID do usuário do agente de publicação/assinatura distribuída no gerenciador de filas remotas. Por exemplo, `QM2` em Figura 30 na página 515. Ao usuário é, então, facilmente concedido acesso aos perfis do objeto do tópico local, porque esse ID do usuário é definido no sistema e, portanto, não haverá nenhum conflito de domínio.

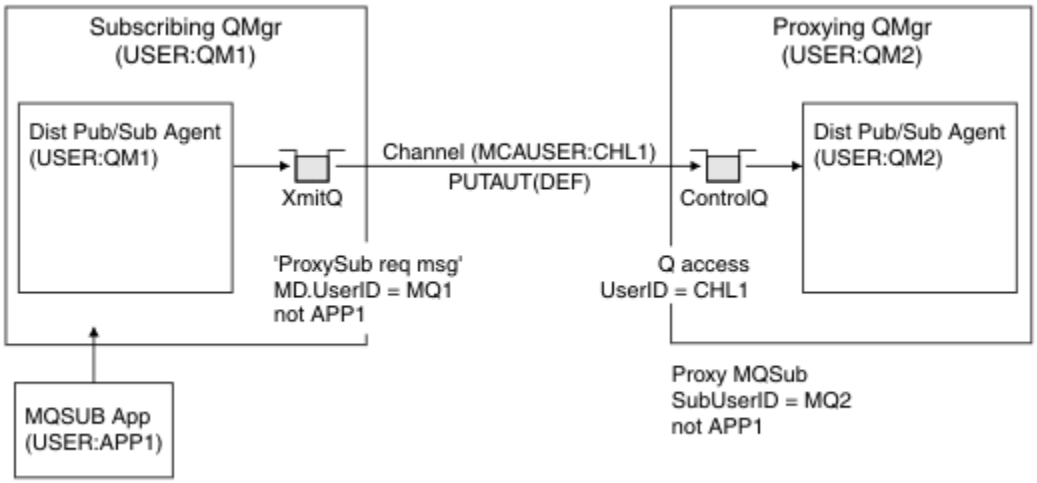


Figura 30. Segurança de Assinatura de Proxy Criando uma Assinatura

Enviando de Volta Publicações Remotas

Quando uma publicação é criada no gerenciador de filas de publicação, uma cópia da publicação é criada para qualquer assinatura de proxy. O contexto da publicação copiada contém o contexto do ID do usuário que fez a assinatura; `QM2` em Figura 31 na página 516. A assinatura de proxy é criada com uma fila de destino que é uma fila remota, portanto, a mensagem de publicação é resolvida em uma fila de transmissão.

A confiança para uma organização para conectar seu gerenciador de filas, QM2, para outro gerenciador de filas, QM1, é confirmada por meios normais de autenticação de canais. Se essa organização confiável tiver permissão para fazer publicação/assinatura distribuída, uma verificação de autoridade será feita quando o canal colocar a mensagem de publicação na fila de publicação de publicação/assinatura distribuída SYSTEM.INTER.QMGR.PUBS. O ID do usuário para a verificação de autoridade da fila depende do valor PUTAUT do canal de recebimento (por exemplo, o ID do usuário do canal, MCAUSER, o contexto da mensagem e outros, dependendo do valor e da plataforma). Para obter mais informações sobre a segurança do canal, consulte Segurança de Canal.

Quando a mensagem de publicação alcançar o gerenciador de filas de assinatura, outro MQPUT para o tópico é feito sob a autoridade desse gerenciador de filas e o contexto com a mensagem é substituído pelo contexto de cada um dos assinantes locais, conforme eles são dados a cada mensagem.

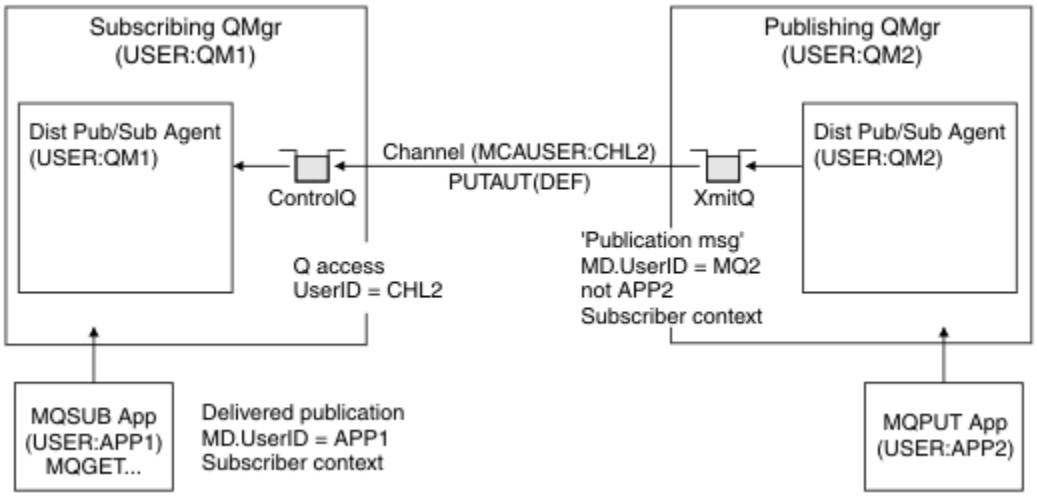


Figura 31. Segurança de Assinatura de Proxy, Publicações de Redirecionamento

Em um sistema em que pouco foi considerado em relação à segurança, os processos de publicação/assinatura distribuídos provavelmente serão executados sob um ID de usuário no grupo mqm, o parâmetro MCAUSER em um canal está em branco (o padrão) e as mensagens são entregues ao várias filas do sistema conforme necessário. O sistema descoberto torna mais fácil de configurar uma prova de conceito para demonstrar a publicação/assinatura distribuída.

Em um sistema no qual a segurança é mais seriamente considerada, essas mensagens internas estão sujeitas aos mesmos controles de segurança que qualquer mensagem vai através do canal.

Se o canal for configurado com um MCAUSER não em branco e um valor PUTAUT especificando que MCAUSER deve ser verificado, então, o MCAUSER em questão deve receber acesso às filas SYSTEM.INTER.QMGR.*. Se houver vários gerenciadores de filas remotas diferentes, com canais em execução em diferentes IDs MCAUSER, todos esses IDs de usuário precisam ter acesso concedido às filas SYSTEM.INTER.QMGR.*. Os canais em execução sob diferentes IDs de MCAUSER pode ocorrer, por exemplo, quando várias conexões hierárquicas forem configuradas em um único gerenciador de filas.

Se o canal estiver configurado com um valor PUTAUT especificando que o contexto da mensagem é usado, o acesso às filas SYSTEM.INTER.QMGR.* será verificado com base no ID do usuário dentro da mensagem interna. Como todas estas mensagens são colocadas com o ID do usuário do agente de publicação/assinatura distribuída a partir do gerenciador de filas que está enviando a mensagem interna ou mensagem de publicação (consulte Figura 31 na página 516), ele não é um conjunto muito grande de IDs do usuário para os quais conceder acesso às diversas filas do sistema (um por gerenciador de filas remotas), se você desejar configurar a segurança de sua publicação/assinatura distribuída desta maneira. Ele ainda tem todos os mesmos problemas que a segurança de contexto de canal sempre possui; a dos domínios de ID do usuário diferente e o fato de que o ID do usuário na mensagem não pode ser definido no sistema receptor. No entanto, é uma maneira perfeitamente aceitável para execução se necessário.

z/OS A Segurança da fila do sistema fornece uma lista de filas e o acesso que é necessário para configurar com segurança do seu ambiente de publicação/assinatura distribuída. Se as mensagens

internas ou quaisquer publicações falharem ao serem colocadas devido a violações de segurança, o canal gravará uma mensagem para o log da maneira normal e as mensagens poderão ser enviadas para a fila de devoluções de acordo com o processamento de erro de canal normal.

Todos os gerenciadores de filas de mensagens para fins de publicação/assinatura distribuída são executados usando a segurança de canal normal.

Para obter informações sobre a restrição de publicações e assinaturas de proxy no nível de tópico, consulte Segurança de publicação/assinatura.

Usando os IDs de Usuário Padrão com uma Hierarquia do Gerenciador de Filas

Se você tiver uma hierarquia de gerenciadores de filas em execução em diferentes plataformas e estiver usando IDs de usuário padrão, observe que esses IDs de usuário padrão diferem entre as plataformas e podem não ser conhecidos na plataforma de destino. Como resultado, um gerenciador de filas em execução em uma plataforma rejeita mensagens recebidas de gerenciadores de filas em outras plataformas com o código de razão `MQRC_NOT_AUTHORIZED`.

Para evitar mensagens sendo rejeitadas, no mínimo, as seguintes autoridades precisam ser incluídas nos IDs de usuário padrão usados em outras plataformas:

- Autoridade *GET *PUT nas filas do SYSTEM.BROKER. filas
- Autoridade *SUB *PUB nos tópicos do SYSTEM.BROKER. tópicos
- Autoridade *ADMCHG *ADMDLT *ADMCRT na fila do SYSTEM.BROKER.CONTROL.QUEUE.

Os IDs de usuário padrão com uma hierarquia do Gerenciador de Filas são os seguintes:

Plataforma	**ID do usuário padrão**
Windows	`mqm`
SistemasAIX and Linux	`mqm`
IBM i	`QMQM`
z/OS	O ID de usuário do espaço de endereço inicializador de canais

Se os gerenciadores de filas em plataformas diferentes de IBM i forem hierarquicamente conectados a um gerenciador de filas em IBM i, crie e conceda acesso ao ID do usuário 'qmqm'.

Se os gerenciadores de filas no IBM i ou z/OS forem conectados hierarquicamente a um gerenciador de filas no AIX, Linux, and Windows, crie e conceda acesso para o ID do usuário 'mqm'.

Se os gerenciadores de filas no Multiplataformas estiverem hierarquicamente conectados a um gerenciador de filas no z/OS, crie e conceda acesso para o ID do usuário do espaço de endereço do inicializador de canais do z/OS

Os IDs de usuário podem ter distinção entre maiúsculas e minúsculas. O gerenciador de filas de origem (se em Multiplataformas) força o ID do usuário a ser todo em maiúsculas. O gerenciador de filas de recebimento (se em AIX, Linux, and Windows) força o ID do usuário a ser todo em minúsculas. Portanto, todos os IDs do usuário criados em sistemas AIX and Linux devem ser criados em sua forma minúscula. Se uma saída de mensagem tiver sido instalada, forçando o ID do usuário em maiúsculas ou minúsculas não ocorrerá. É preciso ter cuidado para entender como a saída de mensagem processa o ID do usuário.

Para evitar problemas em potencial com a conversão de IDs de usuário:

- Nos sistemas AIX, Linux, and Windows, assegure-se de que os IDs do usuário sejam especificados em letra minúscula.
- Em sistemas IBM i e z/OS , assegure-se de que os IDs do usuário sejam especificados em maiúsculas.

IBM MQ Console e a segurança do REST API

A segurança para o IBM MQ Console e o REST API é configurada por meio da edição da configuração do servidor mqweb no arquivo `mqwebuser.xml`.

Sobre esta tarefa

É possível rastrear as ações do usuário e auditar o uso do IBM MQ Console e da REST API examinando os arquivos de log do servidor mqweb.

Os usuários do IBM MQ Console e da REST API podem ser autenticados usando:

- Registro Básico
- registro LDAP
- Registro do S.O. local
- SAF no z/OS
- Qualquer outro tipo de registro suportado pelo WebSphere Liberty

As funções podem ser designadas aos usuários do IBM MQ Console e aos usuários da REST API para determinar qual nível de acesso é concedido a eles para os objetos do IBM MQ. Por exemplo, para executar o sistema de mensagens, os usuários devem ser designados à função `MQWebUser`. Para obter mais informações sobre as funções disponíveis, veja “Funções no IBM MQ Console e na REST API” na página 530.

Depois que um usuário é designado a uma função, há vários métodos que podem ser usados para autenticar o usuário. Com o IBM MQ Console, os usuários podem efetuar login com um nome de usuário e uma senha ou podem usar autenticação por certificado de cliente. Com a REST API, os usuários podem usar a autenticação básica de HTTP, a autenticação baseada em token ou a autenticação por certificado de cliente.

Procedimento

1. Defina o registro do usuário para autenticar usuários e designe a cada usuário ou grupo uma função para autorizar os usuários e grupos a usar o IBM MQ Console ou a REST API. Para obter mais informações, consulte “Configurando usuários e funções” na página 519
2. Escolha como os usuários do IBM MQ Console são autenticados com o servidor mqweb. Você não precisa usar o mesmo método para todos os usuários:
   - Permitir que os usuários se autentifiquem usando a autenticação do token. Neste caso, um usuário insere um ID do usuário e senha no log do IBM MQ Console na tela. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Nenhuma configuração adicional é necessária para usar essa opção de autenticação, mas é possível configurar opcionalmente o tempo de validade para o token LTPA. Para obter mais informações, veja Configurando o intervalo de validação de token LTPA.
   - Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no IBM MQ Console, mas usa o certificado de cliente. Para obter mais informações, consulte “Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console” na página 534.
3. Escolha como os usuários do REST API são autenticados com o servidor mqweb. Você não precisa usar o mesmo método para todos os usuários:
   - Permitir que os usuários se autentifiquem usando a autenticação de HTTP básica. Nesse caso, um nome de usuário e uma senha são codificados, mas não criptografados, e enviados com cada solicitação de REST API para autenticar e autorizar o usuário para essa solicitação. Para que essa autenticação seja segura, deve-se usar uma conexão segura. Ou seja, deve-se usar HTTPS. Para obter mais informações, consulte “Usando autenticação básica HTTP com a REST API” na página 537.

- Permitir que os usuários se autentifiquem usando a autenticação do token. Nesse caso, um usuário fornece um ID de usuário e uma senha para o recurso REST API `login` com o método HTTP POST. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Para obter informações adicionais, consulte "Usando autenticação baseada em token com a API de REST" na página 539.

  Para que essa autenticação seja segura, deve-se usar uma conexão segura. Ou seja, deve-se usar HTTPS. No entanto, se você tiver ativado conexões HTTP, será possível permitir que um token LTPA emitido para uma conexão HTTPS seja usado para uma conexão HTTP. Para obter mais informações, consulte Configurando o token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no REST API, mas usa o certificado de cliente. Para obter mais informações, consulte "Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console" na página 534.

4. Opcional: Configure o Cross Origin Resource Sharing para a REST API.

   Por padrão, um navegador da web não permite scripts, como JavaScript, para chamar a REST API quando o script não é da mesma origem que a REST API. Ou seja, as solicitações de origem cruzada não estão ativadas. É possível configurar o Cross Origin Resource Sharing (CORS) para permitir solicitações de origem cruzada de URLs especificadas. Para obter informações adicionais, consulte "Configurando o CORS para a REST API" na página 541.
5. Opcional: Configure a validação do cabeçalho do host para o IBM MQ Console e a REST API.

   É possível configurar a validação do cabeçalho do host e criar uma lista de permissões de nomes do host e portas para garantir que somente as solicitações que contêm cabeçalhos de host específicos sejam processadas pelo IBM MQ Console e pela REST API. Para obter informações adicionais, consulte "Configurando a validação do cabeçalho do host para o IBM MQ Console e a REST API" na página 542.

Configurando usuários e funções

Para usar o IBM MQ Console ou a REST API, os usuários precisam autenticar-se em um registro do usuário, definido para o servidor mqweb.

Sobre esta tarefa

Os usuários autenticados precisam ser membros de um dos grupos que autorizam o acesso aos recursos do IBM MQ Console e da REST API. Por padrão, o registro do usuário não contém nenhum usuário; estes precisam ser incluindo por meio da edição do arquivo `mqwebuser.xml`.

Ao configurar usuários e grupos, você configura primeiro um registro do usuário para autenticar usuários e grupos. Esse registro do usuário é compartilhado entre o IBM MQ Console e a REST API. É possível controlar se os usuários e grupos têm acesso ao IBM MQ Console, REST API, ou a ambos, ao configurar funções para seus usuários e grupos.

Depois de configurar o registro do usuário, você configura funções para os usuários e grupos para conceder autorização a eles. Há várias funções disponíveis, incluindo funções específicas para usar a REST API para Managed File Transfer. Cada função concede um nível diferente de acesso. Para obter informações adicionais, consulte "Funções no IBM MQ Console e na REST API" na página 530.

Vários arquivos XML de amostra são fornecidos com o servidor mqweb para tornar a configuração de usuários e grupos mais simples. Os usuários que estão familiarizados com a configuração de segurança no WebSphere Liberty (WLP) podem preferir não usar as amostras. O WLP fornece outros recursos de autorização, além daqueles documentados aqui.

Procedimento

- Configure usuários e grupos com um registro básico usando o arquivo `basic_registry.xml`.

Os nomes e as senhas do usuário no registro são utilizados para autenticar e autorizar usuários do IBM MQ Console e da REST API.

Para configurar um registro básico usando o arquivo de amostra `basic_registry.xml`, veja "Configurando um registro básico para o IBM MQ Console e a REST API" na página 521.

- Configure usuários e grupos com um registro LDAP usando o arquivo `ldap_registry.xml`.

  Os nomes de usuário e senhas no registro LDAP são usados para autenticar e autorizar o uso do IBM MQ Console e da REST API.

  Para configurar um registro LDAP usando o arquivo de amostra `ldap_registry.xml`, veja "Configurando um registro LDAP para o IBM MQ Console e a REST API" na página 525.

- ALW

  Configure usuários e grupos com um registro do sistema operacional local usando o arquivo `local_os_registry.xml`.

  Os nomes de usuário e senhas no registro do sistema operacional são usados para autenticar e autorizar os usuários do IBM MQ Console e da REST API.

  Para configurar um registro do S.O. local usando o arquivo de amostra `local_os_registry.xml`, veja "Configurando um registro de S.O. local para o IBM MQ Console e a REST API" na página 524.

- z/OS

  Configure usuários e grupos com a interface System Authorization Facility (SAF) no z/OS usando o arquivo `zos_saf_registry.xml`.

  Os perfis do RACF, ou de outro produto de segurança, são usados para conceder aos usuários e grupos acesso a funções. Os nomes de usuário e senhas no banco de dados RACF são usados para autenticar e autorizar os usuários do IBM MQ Console e da REST API.

  Para configurar a interface SAF usando o arquivo de amostra `zos_saf_registry.xml`, veja "Configuring a SAF registry for the IBM MQ Console and REST API" na página 528.

- Desative a segurança, incluindo a capacidade de acessar o IBM MQ Console ou a REST API usando HTTPS, usando o arquivo `no_security.xml`.

Como proceder a seguir

Escolha como os usuários se autenticam:

Opções de autenticação do IBM MQ Console

- Permitir que os usuários se autentifiquem usando a autenticação do token. Neste caso, um usuário insere um ID do usuário e senha no log do IBM MQ Console na tela. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Nenhuma configuração adicional é necessária para usar essa opção de autenticação, mas é possível configurar opcionalmente o intervalo de validação para o token LTPA. Para obter mais informações, veja Configurando o intervalo de validação de token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no IBM MQ Console, mas usa o certificado de cliente. Para obter mais informações, consulte "Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console" na página 534.

Opções de autenticação do REST API

- Permitir que os usuários se autentifiquem usando a autenticação de HTTP básica. Nesse caso, um nome de usuário e uma senha são codificados, mas não criptografados, e enviados com cada solicitação de REST API para autenticar e autorizar o usuário para essa solicitação. Para que essa autenticação seja segura, deve-se usar uma conexão segura. Ou seja, deve-se usar HTTPS. Para obter mais informações, consulte "Usando autenticação básica HTTP com a REST API" na página 537.

- Permitir que os usuários se autentifiquem usando a autenticação do token. Nesse caso, um usuário fornece um ID de usuário e uma senha para o recurso REST API `login` com o método HTTP POST. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Para obter informações adicionais, consulte "Usando autenticação baseada em token com a API de REST" na página 539. É possível configurar o intervalo de validação para o token LTPA. Para obter mais informações, consulte Configurando o token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no REST API, mas usa o certificado de cliente. Para obter mais informações, consulte "Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console" na página 534.

Configurando um registro básico para o IBM MQ Console e a REST API

É possível configurar um registro básico dentro do arquivo `mqwebuser.xml`. Os nomes de usuário, senhas e funções no arquivo xml são usados para autenticar e autorizar usuários do IBM MQ Console e da REST API.

Antes de começar

- Ao configurar usuários dentro do registro básico, deve-se designar uma função a cada usuário. Cada função fornece diferentes níveis de privilégio para acessar o IBM MQ Console e a REST API, além de determinar o contexto de segurança que será usado quando uma operação permitida for tentada. É necessário entender essas funções antes de configurar o registro básico. Para obter mais informações sobre cada uma das funções, consulte "Funções no IBM MQ Console e na REST API" na página 530.
- Para completar esta tarefa, você deve ser um usuário com privilégios suficientes para editar o arquivo `mqwebuser.xml`:
  - z/OS No z/OS, você deve ter acesso de gravação ao arquivo `mqwebuser.xml`.
  - Multi Em todos os outros sistemas operacionais, deve-se ser um usuário privilegiado.
  - Linux V 9.4.0 Se o servidor mqweb fizer parte de uma instalação independente do IBM MQ Web Server , deve-se ter acesso de gravação ao arquivo `mqwebuser.xml` no diretório de dados IBM MQ Web Server

Procedimento

1. Copie o arquivo XML de amostra `basic_registry.xml` de um dos caminhos a seguir:
   - Em uma instalação do IBM MQ :
     - ALW No AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS No z/OS: `PathPrefix /web/mq/samp/configuration`

       em que `PathPrefix` é o caminho da instalação do IBM MQ for z/OS UNIX System Services Components.
   - Linux V 9.4.0 Em uma instalação IBM MQ Web Server independente: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     em que *MQWEB_INSTALLATION_PATH* é o diretório no qual o arquivo de instalação IBM MQ Web Server foi descompactado.
2. Coloque o arquivo de amostra no diretório apropriado:
   - Em uma instalação do IBM MQ :

– Linux AIX Em AIX ou Linux: `/var/mqm/web/installations/installationName/servers/mqweb`

– Windows Em Windows: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb, em que *MQ_DATA_PATH* é o caminho de dados IBM MQ . Esse caminho é o caminho de dados selecionado durante a instalação do IBM MQ. Por padrão, esse caminho é `C:\ProgramData\IBM\MQ`.

– z/OS No z/OS: *WLP_user_directory*`/servers/mqweb`

em que *WLP_user_directory* é o diretório que foi especificado quando o script **crtmqweb** foi executado para criar a definição do servidor do mqweb.

- Linux V 9.4.0 Em uma instalação IBM MQ Web Server independente: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

  em que *MQ_OVERRIDE_DATA_PATH* é o diretório de dados IBM MQ Web Server para o qual a variável de ambiente do **MQ_OVERRIDE_DATA_PATH** aponta.

3. Opcional: Se você mudou as definições de configuração em `mqwebuser.xml`, copie-as para o arquivo de amostra.
4. Exclua o arquivo `mqwebuser.xml` existente e renomeie o arquivo de amostra para `mqwebuser.xml`.
5. Edite o novo arquivo `mqwebuser.xml` para incluir usuários e grupos dentro das tags **basicRegistry**.

   Esteja ciente de que qualquer usuário com a função `MQWebUser` pode executar somente as operações que o ID do usuário pode executar no gerenciador de filas. Portanto, o ID do usuário definido no registro deve ter um ID do usuário idêntico no sistema no qual o IBM MQ está instalado. Esses IDs de usuários devem estar no mesmo caso ou o mapeamento entre os IDs de usuários pode falhar.

   Para obter mais informações sobre como configurar registros do usuário básico, veja Configurando um registro do usuário básico do Liberty na documentação do WebSphere Liberty.
6. Designe funções a usuários e grupos editando o arquivo `mqwebuser.xml`:

   Há várias funções disponíveis que autorizam usuários e grupos a usarem o IBM MQ Console e o REST API. Cada função concede um nível diferente de acesso. Para obter informações adicionais, consulte “Funções no IBM MQ Console e na REST API” na página 530.

   - Para designar funções e conceder acesso ao IBM MQ Console, inclua seus usuários e grupos entre as tags **security-role** apropriadas dentro das tags **<enterpriseApplication id="com.ibm.mq.console">**.
   - Para designar funções e conceder acesso ao REST API, inclua seus usuários e grupos entre as tags **security-role** apropriadas dentro das tags **<enterpriseApplication id="com.ibm.mq.rest">**.

   Para ajudar com o formato das informações sobre o usuário e grupo dentro das tags **security-role**, veja os exemplos.
7. Se você tiver fornecido senhas aos usuários no `mqwebuser.xml`, será necessário codificá-las para torná-las mais seguras usando o comando **securityUtility encoding** fornecido pelo WebSphere Liberty. Para obter mais informações, consulte o Comando Liberty:securityUtility na documentação do produto WebSphere Liberty.

Exemplo

No exemplo a seguir, é concedido ao grupo `MQWebAdminGroup` acesso ao IBM MQ Console com a função `MQWebAdmin`. O usuário `reader` recebe acesso à função `MQWebAdminRO` e o usuário `guest` recebe acesso à função `MQWebUser`:

```
<enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
            <security-role name="MQWebAdmin">
```

```
                <group name="MQWebAdminGroup" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebAdminRO">
                <user name="reader" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebUser">
                <user name="guest" realm="defaultRealm"/>
            </security-role>
        </application-bnd>
    </enterpriseApplication>
```

No exemplo a seguir, os usuários `reader` e `guest` recebem acesso ao IBM MQ Console. O usuário `user` tem acesso concedido à REST API e quaisquer usuários dentro do grupo `MQAdmin` têm acesso concedido ao IBM MQ Console e à REST API. O usuário `mftadmin` tem acesso concedido à REST API para MFT:

```
<enterpriseApplication id="com.ibm.mq.console">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebAdminRO">
            <user name="reader" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="guest" realm="defaultRealm"/>
        </security-role>
    </application-bnd>
</enterpriseApplication>

<enterpriseApplication id="com.ibm.mq.rest">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQAdmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <user name="user" realm="defaultRealm"/>
        </security-role>
        <security-role name="MFTWebAdmin">
            <user name="mftadmin" realm="defaultRealm"/>
        </security-role>
     </application-bnd>
</enterpriseApplication>
```

Como proceder a seguir

Escolha como os usuários se autenticam:

Opções de autenticação do IBM MQ Console

- Permitir que os usuários se autentifiquem usando a autenticação do token. Neste caso, um usuário insere um ID do usuário e senha no log do IBM MQ Console na tela. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Nenhuma configuração adicional é necessária para usar essa opção de autenticação, mas é possível configurar opcionalmente o intervalo de validação para o token LTPA. Para obter mais informações, veja Configurando o intervalo de validação de token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no IBM MQ Console, mas usa o certificado de cliente. Para obter mais informações, consulte “Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console” na página 534.

Opções de autenticação do REST API

- Permitir que os usuários se autentifiquem usando a autenticação de HTTP básica. Nesse caso, um nome de usuário e uma senha são codificados, mas não criptografados, e enviados com cada solicitação de REST API para autenticar e autorizar o usuário para essa solicitação. Para que essa autenticação seja segura, deve-se usar uma conexão segura. Ou seja, deve-se usar HTTPS. Para obter mais informações, consulte “Usando autenticação básica HTTP com a REST API” na página 537.

- Permitir que os usuários se autentifiquem usando a autenticação do token. Nesse caso, um usuário fornece um ID de usuário e uma senha para o recurso REST API `login` com o método HTTP POST. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Para obter informações adicionais, consulte "Usando autenticação baseada em token com a API de REST" na página 539. É possível configurar o intervalo de validação para o token LTPA. Para obter mais informações, consulte Configurando o token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no REST API, mas usa o certificado de cliente. Para obter mais informações, consulte "Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console" na página 534.

ALW Configurando um registro de S.O. local para o IBM MQ Console e a REST API

É possível configurar um registro do sistema operacional local dentro do arquivo `mqwebuser.xml`. Os nomes de usuário e senhas no sistema operacional local são usados para autenticar e autorizar usuários do IBM MQ Console e da REST API.

Antes de começar

- Para a autenticação por certificado de cliente com o recurso de autenticação do S.O. local, a identidade do usuário é o nome comum (CN) do nome distinto (DN) do certificado de cliente. Se a identidade do usuário não existir como um usuário do sistema operacional, o login de certificado de cliente falhará e efetuará fallback para autenticação baseada em senha.
- Para completar esta tarefa, você deve ser um usuário com privilégios suficientes para editar o arquivo `mqwebuser.xml`:
  - Linux V 9.4.0 Se o servidor mqweb fizer parte de uma instalação independente do IBM MQ Web Server , deve-se ter acesso de gravação ao arquivo `mqwebuser.xml` no diretório de dados IBM MQ Web Server
  - Se o servidor mqweb fizer parte da instalação do IBM MQ , você deverá ser um usuário privilegiado.

Sobre esta tarefa

Com um registro do sistema operacional local, os usuários e grupos são designados automaticamente a uma função:

- Qualquer usuário que faça parte do grupo 'mqm' ou do grupo 'QMQMADM' no IBM irecebe as funções `MQWebAdmin` e `MFTWebAdmin` .
- Todos os outros usuários recebem a função `MQWebUser` .

Para obter informações adicionais sobre essas funções, consulte "Funções no IBM MQ Console e na REST API" na página 530.

Um registro do sistema operacional local pode ser usado somente no AIX, Linux, and Windows.

z/OS A função equivalente é fornecida no z/OS configurando um registro SAF. Para obter mais informações, consulte "Configuring a SAF registry for the IBM MQ Console and REST API" na página 528.

Procedimento

1. Copie o arquivo XML de amostra `local_os_registry.xml` de um dos caminhos a seguir:
   - Linux V 9.4.0 Em uma instalação IBM MQ Web Server independente:
     MQWEB_INSTALLATION_PATH`/web/mq/samp/configuration`

     em que *MQWEB_INSTALLATION_PATH* é o diretório no qual o arquivo de instalação IBM MQ Web Server foi descompactado.

- Em uma IBM MQ instalação: *MQ_INSTALLATION_PATH*/web/mq/samp/configuration

2. Coloque o arquivo de amostra em um dos seguintes diretórios:
   - Linux V 9.4.0 Em uma instalação IBM MQ Web Server independente: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     em que *MQ_OVERRIDE_DATA_PATH* é o diretório de dados IBM MQ Web Server para o qual a variável de ambiente do **MQ_OVERRIDE_DATA_PATH** aponta.
   - Em uma IBM MQ instalação: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb
3. Opcional: Se você mudou as definições de configuração em mqwebuser.xml, copie-as para o arquivo de amostra.
4. Exclua o arquivo mqwebuser.xml existente e renomeie o arquivo de amostra para mqwebuser.xml.

Como proceder a seguir

Escolha como os usuários se autenticam:

Opções de autenticação do IBM MQ Console

- Permitir que os usuários se autentifiquem usando a autenticação do token. Neste caso, um usuário insere um ID do usuário e senha no log do IBM MQ Console na tela. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Nenhuma configuração adicional é necessária para usar essa opção de autenticação, mas é possível configurar opcionalmente o intervalo de validação para o token LTPA. Para obter mais informações, veja Configurando o intervalo de validação de token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no IBM MQ Console, mas usa o certificado de cliente. Para obter mais informações, consulte “Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console” na página 534.

Opções de autenticação do REST API

- Permitir que os usuários se autentifiquem usando a autenticação de HTTP básica. Nesse caso, um nome de usuário e uma senha são codificados, mas não criptografados, e enviados com cada solicitação de REST API para autenticar e autorizar o usuário para essa solicitação. Para que essa autenticação seja segura, deve-se usar uma conexão segura. Ou seja, deve-se usar HTTPS. Para obter mais informações, consulte “Usando autenticação básica HTTP com a REST API” na página 537.
- Permitir que os usuários se autentifiquem usando a autenticação do token. Nesse caso, um usuário fornece um ID de usuário e uma senha para o recurso REST API login com o método HTTP POST. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Para obter informações adicionais, consulte “Usando autenticação baseada em token com a API de REST” na página 539. É possível configurar o intervalo de validação para o token LTPA. Para obter mais informações, consulte Configurando o token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no REST API, mas usa o certificado de cliente. Para obter mais informações, consulte “Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console” na página 534.

Configurando um registro LDAP para o IBM MQ Console e a REST API

É possível configurar um registro LDAP dentro do arquivo mqwebuser.xml. Os nomes de usuário e senhas no registro LDAP são usados para autenticar e autorizar usuários do IBM MQ Console e da REST API.

Antes de começar

- Ao configurar um registro LDAP, deve-se designar uma função a cada usuário. Cada função fornece diferentes níveis de privilégio para acessar o IBM MQ Console e a REST API, além de determinar o contexto de segurança que será usado quando uma operação permitida for tentada. É necessário entender essas funções antes de configurar o registro. Para obter mais informações sobre cada uma das funções, consulte "Funções no IBM MQ Console e na REST API" na página 530.

  Esteja ciente de que qualquer usuário com a função `MQWebUser` pode executar somente as operações que o ID do usuário pode executar no gerenciador de filas. Portanto, o ID do usuário definido no servidor LDAP deve ter um ID do usuário idêntico no sistema no qual o IBM MQ está instalado. Esses IDs de usuários devem estar no mesmo caso ou o mapeamento entre os IDs de usuários pode falhar.
- Para completar esta tarefa, você deve ser um usuário com privilégios suficientes para editar o arquivo `mqwebuser.xml`:
  - z/OS No z/OS, você deve ter acesso de gravação ao arquivo `mqwebuser.xml`.
  - Multi Em todos os outros sistemas operacionais, deve-se ser um usuário privilegiado.
  - Linux V 9.4.0 Se o servidor mqweb fizer parte de uma instalação independente do IBM MQ Web Server , deve-se ter acesso de gravação ao arquivo `mqwebuser.xml` no diretório de dados IBM MQ Web Server

Procedimento

1. Copie o arquivo XML de amostra `ldap_registry.xml` de um dos caminhos a seguir:
   - Em uma instalação do IBM MQ :
     - ALW No AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS No z/OS: `PathPrefix /web/mq/samp/configuration`

       em que `PathPrefix` é o caminho da instalação do IBM MQ for z/OS UNIX System Services Components.
   - Linux V 9.4.0 Em uma instalação IBM MQ Web Server independente: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     em que *MQWEB_INSTALLATION_PATH* é o diretório no qual o arquivo de instalação IBM MQ Web Server foi descompactado.
2. Coloque o arquivo de amostra no diretório apropriado:
   - Em uma instalação do IBM MQ :
     - Linux AIX Em AIX ou Linux: `/var/mqm/web/installations/`*installationName*`/servers/mqweb`
     - Windows Em Windows: *MQ_DATA_PATH*`\web\installations\`*installationName*`\servers\mqweb`, em que *MQ_DATA_PATH* é o caminho de dados IBM MQ . Esse caminho é o caminho de dados selecionado durante a instalação do IBM MQ. Por padrão, esse caminho é `C:\ProgramData\IBM\MQ`.
     - z/OS No z/OS: *WLP_user_directory*`/servers/mqweb`

       em que *WLP_user_directory* é o diretório que foi especificado quando o script **crtmqweb** foi executado para criar a definição do servidor do mqweb.
   - Linux V 9.4.0 Em uma instalação IBM MQ Web Server independente: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

em que *MQ_OVERRIDE_DATA_PATH* é o diretório de dados IBM MQ Web Server para o qual a variável de ambiente do **MQ_OVERRIDE_DATA_PATH** aponta.

3. Opcional: Se você mudou as definições de configuração em `mqwebuser.xml`, copie-as para o arquivo de amostra.
4. Exclua o arquivo `mqwebuser.xml` existente e renomeie o arquivo de amostra para `mqwebuser.xml`.
5. Edite o novo arquivo `mqwebuser.xml` para mudar as configurações de registro LDAP dentro das tags **ldapRegistry** e **idsLdapFilterProperties**.

   Para obter mais informações sobre como configurar registros LDAP, veja Configurando registros do usuário LDAP no Liberty na documentação do WebSphere Liberty.
6. Designe funções a usuários e grupos editando o arquivo `mqwebuser.xml`:

   Há várias funções disponíveis que autorizam usuários e grupos a usarem o IBM MQ Console e o REST API. Cada função concede um nível diferente de acesso. Para obter informações adicionais, consulte "Funções no IBM MQ Console e na REST API" na página 530.

   - Para designar funções e conceder acesso ao IBM MQ Console, inclua seus usuários e grupos entre as tags **security-role** apropriadas dentro das tags **<enterpriseApplication id="com.ibm.mq.console">**.
   - Para designar funções e conceder acesso ao REST API, inclua seus usuários e grupos entre as tags **security-role** apropriadas dentro das tags **<enterpriseApplication id="com.ibm.mq.rest">**.

Como proceder a seguir

Escolha como os usuários se autenticam:

Opções de autenticação do IBM MQ Console

- Permitir que os usuários se autentifiquem usando a autenticação do token. Neste caso, um usuário insere um ID do usuário e senha no log do IBM MQ Console na tela. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Nenhuma configuração adicional é necessária para usar essa opção de autenticação, mas é possível configurar opcionalmente o intervalo de validação para o token LTPA. Para obter mais informações, veja Configurando o intervalo de validação de token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no IBM MQ Console, mas usa o certificado de cliente. Para obter mais informações, consulte "Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console" na página 534.

Opções de autenticação do REST API

- Permitir que os usuários se autentifiquem usando a autenticação de HTTP básica. Nesse caso, um nome de usuário e uma senha são codificados, mas não criptografados, e enviados com cada solicitação de REST API para autenticar e autorizar o usuário para essa solicitação. Para que essa autenticação seja segura, deve-se usar uma conexão segura. Ou seja, deve-se usar HTTPS. Para obter mais informações, consulte "Usando autenticação básica HTTP com a REST API" na página 537.
- Permitir que os usuários se autentifiquem usando a autenticação do token. Nesse caso, um usuário fornece um ID de usuário e uma senha para o recurso REST API `login` com o método HTTP POST. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Para obter informações adicionais, consulte "Usando autenticação baseada em token com a API de REST" na página 539. É possível configurar o intervalo de validação para o token LTPA. Para obter mais informações, consulte Configurando o token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no REST API, mas usa o certificado de cliente. Para obter mais informações, consulte "Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console" na página 534.

z/OS Configuring a SAF registry for the IBM MQ Console and REST API

The System Authorization Facility (SAF) interface allows the mqweb server to call the external security manager for authentication and authorization checking. A user can then log in to the IBM MQ Console and REST API with a z/OS user ID and password.

Before you begin

- When you configure a SAF registry, you must assign users a role. Each role provides different levels of privilege to access the IBM MQ Console and REST API, and determines the security context that is used when an allowed operation is attempted. You need to understand these roles before you configure the registry. For more information about each of the roles, see "Funções no IBM MQ Console e na REST API" on page 530.
- You need the WebSphere Liberty Angel process running to use the authorized interface to SAF. See Enabling z/OS authorized services on Liberty for z/OS for more information.
- To complete this task, you must have write access to the `mqwebuser.xml` file, and authority to define security manager profiles.

Note: From IBM MQ 9.3.5 for Continuous Delivery and from IBM MQ 9.3.0 Fix Pack 20 for Long Term Support, the sample configuration file `zos_saf_registry.xml` is updated to remove a duplicate safAuthorization entry.

This update fixes an issue where an ICH408I error can occur when the IBM MQ Console on z/OS is upgraded to a level that ships WebSphere Liberty Profile 22.0.0.12 or later: that is, from IBM MQ 9.3.0 Fix Pack 2 for Long Term Support and from IBM MQ 9.3.1 CSU 1 and IBM MQ 9.3.2 for Continuous Delivery. Having more than one safAuthorization statement is not supported and might cause an ICH408I error when users who are not in either MQWebAdmin or MQWebAdminRO roles, in the EBJROLE class, try to access a z/OS queue manager through the IBM MQ Console.

The default for **racRouteLog**, which specifies the types of access attempts to log, is NONE. If you require an additional report or record for security auditing, see SAF Authorization (safAuthorization) for more information.

About this task

The SAF interface allows the mqweb server to call the external security manager for authentication and authorization checking for both the IBM MQ Console and REST API.

Procedure

1. Follow the steps in Enabling z/OS authorized services on Liberty for z/OS to give your mqweb server access to use z/OS authorized services.

   Sample JCL for starting the angel process is in USS_ROOT/web/templates/zos/procs/bbgzangl.jcl, where USS_ROOT is the path in z/OS UNIX System Services (z/OS UNIX) where z/OS UNIX components are installed.

   In bbgzangl.jcl, change the SET ROOT statement to point to USS_ROOT/web, for example, /usr/lpp/mqm/V9R2M0/web.

   See Administering Liberty on z/OS for further information on stopping and starting the angel process.
2. Follow the steps in Liberty: Setting up the System Authorization Facility (SAF) unauthenticated user to create the unauthenticated user needed by Liberty.
3. Copy the `zos_saf_registry.xml` file from the following path: `PathPrefix /web/mq/samp/configuration` where `PathPrefix` is the z/OS UNIX Components installation path.
4. Place the sample file in the *WLP_user_directory*`/servers/mqweb` directory, where *WLP_user_directory* is the directory that was specified when the **crtmqweb** script ran to create the mqweb server definition.

5. Optional: If you previously changed any configuration settings in `mqwebuser.xml`, copy them into the sample file.
6. Delete the existing `mqwebuser.xml` file and rename the sample file to `mqwebuser.xml`.
7. Customize the **safCredentials** element in `mqwebuser.xml`.
   a. Set **profilePrefix** to a name that is unique to your Liberty server. If you have more than one mqweb server running on a single system, you will need to choose a different name for each server; for example MQWEB920 and MQWEB915.
   b. Set **unauthenticatedUser** to the name of the unauthenticated user created in step "2" on page 528.
8. Define the mqweb server APPLID to RACF.

   The APPLID resource name is the value you specified in the **profilePrefix** attribute in step "7" on page 529. The following example defines the mqweb server APPLID in RACF:

   ```
   RDEFINE APPL profilePrefix UACC(NONE)
   ```
9. Grant all users, or groups, to be authenticated to the IBM MQ Console or REST API READ access to the mqweb server APPLID in the APPL class.

   You must also do this for the unauthenticated user defined in step "2" on page 528. The following example grants a user READ access to the mqweb server APPLID in RACF:

   ```
   PERMIT profilePrefix CLASS(APPL) ACCESS(READ) ID(userID)
   ```
10. Use the **SETROPTS** RACF command to refresh the in-storage RACLISTed APPL class profiles:

    ```
    SETROPTS RACLIST(APPL) REFRESH
    ```
11. Define the profiles in the EJBROLE class needed to give users access to roles in the IBM MQ Console and REST API.

    The following example defines the profiles in RACF, where **profilePrefix** is the value specified for the **profilePrefix** attribute in step "7" on page 529.

```
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.console.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebAdminRO UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MQWebUser UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdmin UACC(NONE)
RDEFINE EJBROLE profilePrefix.com.ibm.mq.rest.MFTWebAdminRO UACC(NONE)
```

12. Grant users access to roles in the IBM MQ Console and REST API.

    To do this, give users or groups READ access to one or more of the profiles in the EBJROLE class created in step "11" on page 529. For more information about the roles, see "Funções no IBM MQ Console e na REST API" on page 530.

    The following example gives a user access to the MQWebAdmin role for the REST API in RACF, where **profilePrefix** is the value specified for the **profilePrefix** attribute in step "7" on page 529.

```
PERMIT profilePrefix.com.ibm.mq.rest.MQWebAdmin CLASS(EJBROLE) ACCESS(READ) ID(userID)
```

Results

You have set up SAF authentication for the IBM MQ Console and REST API.

What to do next

Escolha como os usuários se autenticam:

Opções de autenticação do IBM MQ Console

- Permitir que os usuários se autentifiquem usando a autenticação do token. Neste caso, um usuário insere um ID do usuário e senha no log do IBM MQ Console na tela. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo

configurado. Nenhuma configuração adicional é necessária para usar essa opção de autenticação, mas é possível configurar opcionalmente o intervalo de validação para o token LTPA. Para obter mais informações, veja Configurando o intervalo de validação de token LTPA.

- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no IBM MQ Console, mas usa o certificado de cliente. Para obter mais informações, consulte “Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console” on page 534.

Opções de autenticação do REST API

- Permitir que os usuários se autentifiquem usando a autenticação de HTTP básica. Nesse caso, um nome de usuário e uma senha são codificados, mas não criptografados, e enviados com cada solicitação de REST API para autenticar e autorizar o usuário para essa solicitação. Para que essa autenticação seja segura, deve-se usar uma conexão segura. Ou seja, deve-se usar HTTPS. Para obter mais informações, consulte “Usando autenticação básica HTTP com a REST API” on page 537.
- Permitir que os usuários se autentifiquem usando a autenticação do token. Nesse caso, um usuário fornece um ID de usuário e uma senha para o recurso REST API `login` com o método HTTP POST. É gerado um token LTPA que permite que o usuário permaneça com login efetuado e autorizado por um período de tempo configurado. Para obter informações adicionais, consulte “Usando autenticação baseada em token com a API de REST” on page 539. É possível configurar o intervalo de validação para o token LTPA. Para obter mais informações, consulte Configurando o token LTPA.
- Permitir que os usuários se autentifiquem usando certificados de cliente. Nesse caso, o usuário não usa um ID do usuário ou senha para efetuar login no REST API, mas usa o certificado de cliente. Para obter mais informações, consulte “Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console” on page 534.

Funções no IBM MQ Console e na REST API

Ao autorizar usuários e grupos a usar IBM MQ Console ou REST API, deve-se atribuir aos usuários e grupos uma das funções disponíveis: **MQWebAdmin**, **MQWebAdminRO**, **MQWebUser**, **MFTWebAdmin** e **MFTWebAdminRO**. Cada função fornece diferentes níveis de privilégio para acessar o IBM MQ Console e a REST API, além de determinar o contexto de segurança que será usado quando uma operação permitida for tentada.

Nota: Com exceção da função **MQWebUser**, o ID do usuário não faz distinção entre maiúsculas de minúsculas. Consulte “MQWebUser” na página 531 para os requisitos específicos para esta função.

MQWebAdmin

Um usuário ou grupo que é designado a essa função pode executar todas as operações administrativas e opera sob o contexto de segurança do ID do usuário do sistema operacional que é usado para iniciar o servidor mqweb.

Um usuário ou grupo com essa função não tem acesso aos serviços REST a seguir:

- O REST API para MFT. Para utilizar esses serviços, o usuário ou grupo também deve ser designado a função **MFTWebAdmin** ou **MFTWebAdminRO**.
- O messaging REST API. Para usar o messaging REST API, o usuário deve ser designado à função **MQWebUser**.

MQWebAdminRO

Essa função fornece acesso somente leitura para o IBM MQ Console ou a REST API. Um usuário ou um grupo ao qual essa função está designada pode executar as operações a seguir:

- Exibir e consultar operações nos objetos do IBM MQ, como filas e canais.
- Procurar mensagens em filas.

Um usuário ou um grupo ao qual essa função está designada opera no contexto de segurança do ID de usuário do sistema operacional usado para iniciar o servidor mqweb.

Um usuário ou grupo com essa função não tem acesso aos serviços REST a seguir:

- O REST API para MFT. Para utilizar esses serviços, o usuário ou grupo também deve ser designado a função **MFTWebAdmin** ou **MFTWebAdminRO**.
- O messaging REST API. Para usar o messaging REST API, o usuário deve ser designado à função **MQWebUser**.

MQWebUser
Um usuário ou grupo que é designado a essa função pode executar qualquer operação que o ID do usuário pode executar no gerenciador de filas. Por exemplo:

- Operações de início e parada em objetos IBM MQ , tais como canais.
- Definir e configurar operações nos objetos do IBM MQ, como filas e canais.
- Exibir e consultar operações nos objetos do IBM MQ, como filas e canais.
- Colocar e obter mensagens usando o messaging REST API.

Um usuário ou um grupo ao qual essa função está designada opera no contexto de segurança do principal e pode executar somente as operações que o ID do usuário pode executar no gerenciador de filas.

Portanto, o usuário ou grupo que está definido no registro do usuário mqweb deve receber autoridade dentro do IBM MQ para que esse usuário possa executar quaisquer operações. Ao utilizar esta função, é possível controlar precisamente quais usuários possuem qual tipo de acesso a recursos específicos do IBM MQ quando eles usam o IBM MQ Console e o REST API.

Nota:

- O comprimento máximo de um ID do usuário que é designado a essa função é 12 caracteres.
- O caso do ID do usuário deve ser o mesmo no registro do usuário mqweb e no sistema IBM MQ. Se as maiúsculas e minúsculas do ID do usuário forem diferentes, o usuário poderá ser autenticado pelo IBM MQ Console e pelo REST API, mas não autorizado a usar os recursos do IBM MQ.

MFTWebAdmin
Um usuário ou grupo atribuído a essa função pode executar todas as operações MFT de REST e opera sob o contexto de segurança do ID do usuário do sistema operacional usado para iniciar o servidor `mqweb`.

Um usuário ou grupo com essa função não tem acesso a nenhum dos serviços do IBM MQ REST API. Para utilizar esses serviços, o usuário ou grupo também deve ser designado à função **MQWebAdmin**, **MQWebAdminRO** ou **MQWebUser**.

MFTWebAdminRO
Essa função fornece acesso somente leitura à REST API para MFT. Um usuário ou grupo ao qual foi designada essa função pode executar operações somente leitura (solicitações GET) como transferência e agentes de lista.

Um usuário ou um grupo ao qual essa função está designada opera no contexto de segurança do ID de usuário do sistema operacional usado para iniciar o servidor mqweb.

Um usuário ou grupo com essa função não tem acesso a nenhum dos serviços do IBM MQ REST API. Para utilizar esses serviços, o usuário ou grupo também deve ser designado à função **MQWebAdmin**, **MQWebAdminRO** ou **MQWebUser**.

Para obter mais informações sobre como configurar usuários e grupos para usar essas funções, veja “Configurando usuários e funções” na página 519.

Funções de sobreposição

Mais de uma função pode ser designada a um usuário ou a um grupo. Quando um usuário executa uma operação nessa situação, a função com privilégio mais alto que for aplicável à operação será usada. Por exemplo, se um usuário com as funções **MQWebAdminRO** e **MQWebUser** executa uma operação de fila de consulta, a função **MQWebAdminRO** é usada e a operação é tentada no contexto do ID do usuário do

sistema que iniciou o servidor web. Se esse mesmo usuário executar uma operação de definição, a função **MQWebUser** será usada e a operação será tentada no contexto do principal.

ALW Alterando o certificado apresentado pelo IBM MQ Console para seu navegador

É possível configurar o IBM MQ Console para apresentar um certificado assinado por CA para propósitos de autenticação. Se você configurar o IBM MQ Console para apresentar um certificado assinado por CA, o navegador não apresentará mais o aviso de certificado autoassinado quando o IBM MQ Console for acessado.

Sobre esta tarefa

A segurança para o IBM MQ Console é fornecida pelo servidor mqweb que executa o IBM MQ Console Para mudar o certificado que o servidor mqweb apresenta para o seu navegador, primeiro inclua o novo certificado no keystore do servidor mqweb Em seguida, edite a configuração de segurança no arquivo `mqwebuser.xml` para especificar o certificado que o servidor apresenta

O procedimento faz as seguintes suposições:

- Você é um usuário privilegiado .
- Você está usando um sistema AIX, Linuxou Windows .
- Que seu arquivo `mqwebuser.xml` é baseado nos arquivos XML de amostra `basic_registry.xml`, `local_os_registry.xml`ou `ldap_registry.xml`

Procedimento

1. Opcional: Mude a senha padrão do keystore do servidor mqweb `key.jks` usando o comando **runmqktool** :

   ```
   runmqktool -storepasswd -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
   servers/mqweb/resources/security/key.jks -storepass oldPassword
             -new newPassword
   ```

   oldPassword
   Especifica a senha existente do `key.jks` A senha padrão é `password`.

   newPassword
   Especifica uma nova senha do `key.jks`

2. Crie um par de chaves e uma solicitação de certificado para enviar à autoridade de certificação:
   a) Crie o par de chaves usando o comando **runmqktool** :

   ```
   runmqktool -genkeypair -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
   servers/mqweb/resources/security/key.jks -storepass password -storetype JKS
              -alias label -dname distinguished_name
              -sigalg signature_algorithm
   ```

   senha
   Especifica a senha do keystore `key.jks` .

   rótulo
   Especifica o rótulo certificado. Por exemplo, `MQWebConsole`.

   distinguished_name
   Especifica o Nome distinto X.500 para o certificado. Coloque o Nome Distinto entre aspas duplas

   Por exemplo,`"cn=MQWebConsole,o=myOrg,c=UK"`

   signature_algorithm
   Especifica o algoritmo a ser usado para assinar o certificado Para obter mais informações, consulte Algoritmos de assinatura.

b) Crie a solicitação de certificado usando o comando **runmqktool** :

```
runmqktool -certreq -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password -alias label
           -file filename
```

senha
Especifica a senha do keystore key.jks .

rótulo
Especifica o rótulo certificado da subetapa “2.a” na página 532.

filename
Especifica o nome completo do arquivo para a solicitação de certificado

3. Envie o arquivo de solicitação de certificado para uma autoridade de certificação (CA).
4. Quando você tiver o certificado da CA, importe o certificado e quaisquer outros certificados na cadeia de certificados, começando com o certificado da CA raiz, para o keystore keys.jks usando o comando **runmqktool** :

```
runmqktool -importcert -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password
          -alias label -file filename
```

senha
Especifica a senha do keystore key.jks .

rótulo
Especifica o rótulo certificado da subetapa “2.a” na página 532.

filename
Especifica o nome completo do arquivo do certificado a ser importado

5. Configure o servidor mqweb para apresentar o certificado CA:
   a) Abra o arquivo mqwebuser.xml.

   O arquivo mqwebuser.xml pode ser localizado no caminho a seguir: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

   b) Desative a configuração de segurança padrão comentando a linha a seguir:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   Se você configurou o servidor mqweb para usar a autenticação por certificado de cliente, esta linha do arquivo xml já está comentada.

   c) Remova o comentário da seção no arquivo mqwebuser.xml que ativa a configuração de certificado customizado A seção contém o texto a seguir:

```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
    <keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
    <ssl id="thisSSLConfig" clientAuthenticationSupported="true"
keyStoreRef="defaultKeyStore"
           trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"
serverKeyAlias="default"/>
    <sslDefault sslRef="thisSSLConfig"/>
```

   Se você configurou o servidor mqweb para usar a autenticação por certificado de cliente, esta seção do arquivo xml já terá seus comentários cancelado..

   d) Opcional: Se você mudou a senha para o keystore key.jks na etapa “1” na página 532, mude o valor para **password** nas tags defaultKeyStore para uma versão codificada da senha que você configurou:

      i) No diretório *MQ_INSTALLATION_PATH*/web/bin , insira o comando a seguir:

```
securityUtility encode password
```

      ii) Coloque a saída desse comando no campo **password** para o defaultKeyStore.

e) Se você não estiver usando a autenticação por certificado de cliente, comente a linha a seguir:

```
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
```

f) Altere o valor de **serverKeyAlias** de `default` para o valor do rótulo do certificado CA.

6. Pare o servidor mqweb usando o comando **endmqweb** ..
7. Inicie o servidor mqweb usando o comando **strmqweb** ..

Resultados

Quando o servidor da web for iniciado, navegue até o seu IBM MQ Console e atualize. O certificado de CA é usado e você é levado diretamente para a página de login.

ALW Configurando a autenticação por certificado de cliente com o REST API e IBM MQ Console

É possível mapear certificados de cliente para principais para autenticar usuários do IBM MQ Console e da REST API.

Antes de começar

- Configure usuários, grupos e funções para que sejam autorizados a usar o IBM MQ Console e a REST API. Para obter informações adicionais, consulte “Configurando usuários e funções” na página 519.
- Quando você usa a REST API, é possível consultar as credenciais do usuário atual utilizando o método HTTP GET no recurso `login`, fornecendo o certificado de cliente para autenticar a solicitação. Essa solicitação retorna informações sobre o nome do usuário, e as funções às quais o usuário está designado Para obter mais informações, consulte GET `/login`.
- Ao mapear certificados de cliente para principais para autenticar usuários, o nome distinto do certificado de cliente é usado para corresponder a usuários no registro de usuário configurado:
  - Para um registro básico, o Nome Comum (CN) é correspondido ao usuário. Por exemplo, `CN=Fred, O=IBM, C=GB` é correspondido com um nome de usuário de `Fred`.
  - Para um registro LDAP, por padrão, o nome distinto completo é correspondido ao LDAP. É possível configurar filtros e mapeamentos para customizar a correspondência.. Para obter mais informações, veja Modo de mapa de certificado Liberty:LDAP na documentação do WebSphere Liberty.

Sobre esta tarefa

Quando um usuário é autenticado usando um certificado de cliente, o certificado é usado no lugar de um nome de usuário e uma senha. Para a REST API, o certificado de cliente é fornecido com cada solicitação REST para autenticar o usuário. Para o IBM MQ Console, quando um usuário efetua login com um certificado, não é possível efetuar seu logout em seguida.

ALW Em sistemas AIX, Linuxou Windows , o procedimento assume as seguintes informações:

- Que seu arquivo `mqwebuser.xml` é baseado nos arquivos XML de amostra `basic_registry.xml`, `local_os_registry.xml`ou `ldap_registry.xml`
- Que você é um usuário privilegiado

z/OS Para configurar a autenticação por certificado de cliente com um conjunto de chaves do RACF em sistemas z/OS , siga o procedimento em “Configuring TLS for the REST API and IBM MQ Console on z/OS” na página 546

Nota: O procedimento a seguir descreve as etapas necessárias para usar certificados de cliente com o IBM MQ Console e a REST API. Para conveniência do desenvolvedor, as etapas detalham como criar e usar certificados autoassinados. No entanto, para produção, use certificados que são obtidos de uma autoridade de certificação.

Procedimento

1. Crie um certificado usando o comando **runmqktool** :

```
runmqktool -genkeypair -keystore filename -storepass password -storetype PKCS12
           -alias label -dname distinguished_name
           -sigalg signature_algorithm
```

filename
Especifica o nome do keystore, por exemplo, `user.p12`. Se o keystore não existir, ele será criado quando o comando for executado.

senha
Especifica a senha do keystore.

rótulo
Especifica o rótulo certificado. Por exemplo, `user1`.

distinguished_name
Especifica o Nome distinto X.500 para o certificado. Coloque o Nome Distinto entre aspas duplas

Se você estiver usando um registro do usuário básico, insira o nome de um usuário de seu registro do usuário na parte do Nome Comum (CN) do Nome Distinto. Por exemplo, para um usuário `mqadmin`, use o Nome Distinto `"CN=mqadmin"`

Se você estiver usando um registro do S.O. local, insira o nome de um ID do usuário do S.O. local na parte Nome Comum (CN) do Nome Distinto. Por exemplo, para um usuário `mqadmin`, use o Nome Distinto `"CN=mqadmin"`

Se você estiver usando um registro do usuário LDAP, insira um Nome distinto que corresponda ao Nome distinto no registro LDAP.

signature_algorithm
Especifica o algoritmo a ser usado para assinar o certificado Para obter mais informações, consulte Algoritmos de assinatura.

2. Opcional: Obter um certificado de uma autoridade de certificação (CA). Como alternativa, para usar um certificado autoassinado, continue a etapa “3” na página 536.

a) Para obter um certificado de uma autoridade de certificação, crie uma solicitação de certificado usando o comando **runmqktool** :

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

filename
Especifica o nome do keystore da etapa “1” na página 535.

senha
Especifica a senha do keystore.

rótulo
Especifica o rótulo certificado da etapa “1” na página 535.

filename
Especifica o nome completo do arquivo para a solicitação de certificado

b) Envie o arquivo de solicitação de certificado para uma autoridade de certificação (CA).

c) Quando você tiver o certificado da CA, importe o certificado em seu keystore usando o comando **runmqktool** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

filename
Especifica o nome do keystore da etapa “1” na página 535.

senha
Especifica a senha do keystore.

rótulo
Especifica o rótulo certificado da etapa “1” na página 535.

filename
Especifica o nome completo do arquivo do certificado de autoridade de certificação.

3. Extraia a parte pública do certificado usando o comando **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass password
           -alias label -file filename -rfc
```

filename
Especifica o nome do keystore da etapa “1” na página 535.

senha
Especifica a senha do keystore.

rótulo
Especifica o rótulo certificado da etapa “1” na página 535.

filename
Especifica o nome completo do arquivo para o certificado extraído

4. Importe a parte pública do certificado para o keystore de confiança do servidor mqweb como um certificado de assinante para que o servidor possa validar o certificado de cliente usando o comando **runmqktool** :

```
runmqktool -importcert -keystore  MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/trust.jks -storepass password
           -alias label -file filename
```

senha
Especifica a senha do keystore `trust.jks` . É possível especificar uma senha para um keystore `trust.jks` existente ou uma nova senha para um novo keystore `trust.jks` .

rótulo
Especifica o rótulo certificado da etapa “1” na página 535.

filename
Especifica o nome completo do arquivo do certificado extraído

5. Configure o servidor mqweb para usar a autenticação por certificado de cliente:
   a) Abra o arquivo `mqwebuser.xml`.

      O arquivo `mqwebuser.xml` pode ser localizado no caminho a seguir: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`

   b) Desative a configuração de segurança padrão comentando a linha a seguir:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

      Se você configurou o servidor mqweb para apresentar um certificado de autoridade de certificação para o navegador, essa linha já será comentada.

   c) Descomente a seção no arquivo `mqwebuser.xml` que ativa a autenticação de certificado de cliente. A seção contém o texto a seguir:

```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
    <keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
    <ssl id="thisSSLConfig" clientAuthenticationSupported="true"
keyStoreRef="defaultKeyStore"
           trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"
serverKeyAlias="default"/>
    <sslDefault sslRef="thisSSLConfig"/>
```

      Se você configurou o servidor mqweb para apresentar um certificado de CA para o navegador, esta seção já terá o comentário removido. No entanto, pode ser necessário remover o comentário da linha **defaultTrustStore** ..

d) Mude o valor de **password** para o `defaultTrustStore` para corresponder à senha do keystore `trust.jks`:

i) No diretório *MQ_INSTALLATION_PATH*`/web/bin` , insira o comando a seguir:

```
securityUtility encode password
```

ii) Coloque a saída desse comando no campo **password** para o `defaultTrustStore`.

6. Pare o servidor mqweb usando o comando **endmqweb** ..
7. Inicie o servidor mqweb usando o comando **strmqweb** ..
8. Use o certificado de cliente para autenticar:
   - Para usar o certificado de cliente com o IBM MQ Console, instale o certificado de cliente no navegador da web que é usado para acessar o IBM MQ Console.
   - Para usar o certificado de cliente com a REST API, forneça o certificado de cliente com cada solicitação REST. Ao usar os métodos HTTP POST, PATCH ou DELETE, deve-se fornecer autenticação extra com o certificado de cliente para evitar ataques de falsificação de solicitação entre sites. Ou seja, a autenticação extra é usada para confirmar que as credenciais que estão sendo usadas para autenticar a solicitação estão sendo usadas pelo proprietário das credenciais.

     Essa autenticação extra é fornecida pelo cabeçalho HTTP `ibm-mq-rest-csrf-token`. Configure o valor do cabeçalho `ibm-mq-csrf-token` para qualquer coisa, incluindo em branco, em seguida, envie a solicitação.

Exemplo

Importante: No exemplo, nem todas as implementações cURL suportam certificados autoassinados, portanto, deve-se usar uma implementação cURL que o faça.

O exemplo de cURL a seguir mostra como criar uma nova fila `Q1`, em um gerenciador de filas `QM1`, com autenticação por certificado cliente. A configuração exata desse comando cURL depende das bibliotecas com as quais o cURL foi construído O exemplo é baseado em um sistema Windows com cURL construído em OpenSSL.

- Use o método HTTP POST com o recurso de fila, autenticando com o certificado de cliente e incluindo o cabeçalho de HTTP `ibm-mq-rest-csrf-token` com um valor arbitrário. Esse valor pode ser qualquer coisa, incluindo em branco. A sinalização `--cert-type` especifica que o certificado é um certificado PKCS#12. A sinalização `--cert` especifica o local do certificado, seguido por dois pontos e, em seguida, a senha para o certificado:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -
-cert-type P12 --cert c:\user.p12:password
-H "ibm-mq-rest-csrf-token: value"
-H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
```

Usando autenticação básica HTTP com a REST API

Os usuários da REST API podem se autenticar fornecendo seus IDs e senhas em um cabeçalho de HTTP. Para usar esse método de autenticação com métodos de HTTP, como POST, PATCH e DELETE, o cabeçalho de HTTP `ibm-mq-rest-csrf-token` também deve ser fornecido, bem como o ID do usuário e a senha.

Antes de começar

- Configure os usuários, os grupos e as funções para que sejam autorizados a usar a REST API. Para obter mais informações, consulte “Configurando usuários e funções” na página 519.

- Assegure-se de que a autenticação básica HTTP esteja ativada. Verifique se o XML a seguir está presente e não está comentado no arquivo `mqwebuser.xml`. Este XML deve estar dentro das tags `<featureManager>`:

  ```
  <feature>basicAuthenticationMQ-1.0</feature>
  ```

  z/OS No z/OS, deve-se ser um usuário que tenha acesso de gravação ao `mqwebuser.xml` para editar este arquivo.

  Multi Em todos os outros sistemas operacionais, você deve ser um usuário privilegiado para editar o arquivo `mqwebuser.xml`.
- Assegure-se de que você esteja usando uma conexão segura ao enviar solicitações REST. Como a combinação de nome de usuário e senha é codificada, mas não criptografada, deve-se usar uma conexão segura (HTTPS) ao utilizar a autenticação básica HTTP com a REST API.
- É possível consultar as credenciais do usuário atual usando o método HTTP GET no recurso `login`, fornecendo as informações básicas sobre autenticação para autenticar a solicitação. Esta solicitação retorna informações sobre o nome do usuário e as funções designadas ao usuário. Para obter mais informações, consulte GET `/login`.

Procedimento

1. Concatene o nome do usuário com dois-pontos e a senha. Observe que o nome do usuário faz distinção entre maiúsculas e minúsculas.

   Por exemplo, um nome de usuário de administrador e uma senha de administrador tornam-se a sequência a seguir:

   ```
   admin:admin
   ```
2. Codifique essa sequência de nome do usuário e senha na codificação base64.
3. Inclua esse nome do usuário e senha codificados em um cabeçalho de HTTP `Authorization: Basic`.

   Por exemplo, com um nome de usuário codificado de administrador e uma senha de administrador, o cabeçalho a seguir é criado:

   ```
   Authorization: Basic YWRtaW46YWRtaW4=
   ```
4. Quando você usa os métodos de HTTP POST, PATCH ou DELETE, deve-se fornecer autenticação extra, bem como um nome de usuário e uma senha.

   Essa autenticação extra é fornecida pelo cabeçalho HTTP `ibm-mq-rest-csrf-token`. O cabeçalho de HTTP `ibm-mq-rest-csrf-token` deve estar presente na solicitação, mas seu valor pode ser qualquer coisa, incluindo em branco.
5. Envie sua solicitação REST para o IBM MQ com os cabeçalhos apropriados.

Exemplo

O exemplo a seguir mostra como criar uma nova fila Q1, no gerenciador de filas QM1, com autenticação básica em sistemas Windows. O exemplo usa o cURL:

- Use o método de HTTP POST com o recurso de fila, autenticando com a autenticação básica e incluindo o cabeçalho de HTTP `ibm-mq-rest-csrf-token` com um valor arbitrário. Esse valor pode ser qualquer coisa, incluindo em branco:

  ```
  curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST
  -u mqadmin:mqadmin
  -H "ibm-mq-rest-csrf-token: value"
  -H "Content-Type: application/json" --data "{\"name\":\"Q1\"}"
  ```

Usando autenticação baseada em token com a API de REST

Os usuários do REST API podem autenticar fornecendo um ID do usuário e senha para o recurso REST API `login` com o método HTTP POST. Um token LTPA é gerado, que permite que o usuário autentique solicitações futuras. Esse token LTPA tem o prefixo `LtpaToken2`. O usuário pode efetuar logout usando o método HTTP DELETE e pode consultar as informações de login do usuário atual com o método HTTP GET.

Antes de começar

- Configure os usuários, os grupos e as funções para que sejam autorizados a usar a REST API. Para obter mais informações, consulte "Configurando usuários e funções" na página 519.
- Por padrão, o nome do cookie que inclui o token LTPA inicia com `LtpaToken2` e inclui um sufixo que pode ser mudado quando o servidor mqweb é reiniciado. Esse nome de cookie escolhido a esmo permite que mais de um servidor do mqweb seja executado no mesmo sistema. No entanto, se você desejar que o nome do cookie permaneça um valor consistente, será possível especificar o nome dele usando o comando **setmqweb**. Para obter mais informações, consulte Configurando o token LTPA.
- Por padrão, o cookie do token LTPA expira após 120 minutos. É possível configurar o tempo de validade do cookie de token LTPA usando o comando **setmqweb**. Para obter mais informações, consulte Configurando o token LTPA.
- Assegure-se de que você esteja usando uma conexão segura ao enviar solicitações REST. Quando você usa o método HTTP POST no recurso `login`, a combinação de nome do usuário e senha enviada com a solicitação não é criptografada. Portanto, deve-se usar uma conexão segura (HTTPS) ao usar a autenticação baseada em token com a REST API. Por padrão, não é possível usar HTTP com a autenticação do token LTPA. É possível ativar o token LTPA para ser usado por conexões HTTP não seguras, configurando **secureLTPA** como `False`. Para obter mais informações, consulte Configurando o token LTPA.
- É possível consultar as credenciais do usuário atual usando o método de HTTP GET no recurso `login`, fornecendo o token LTPA para autenticar a solicitação. Esta solicitação retorna informações sobre o nome do usuário e as funções designadas ao usuário. Para obter mais informações, consulte GET / `login`.

Procedimento

1. Efetue login de um usuário:
   a) Use o método HTTP POST no recurso `login`:

   ```
   https://host:port/ibmmq/rest/v1/login
   ```

   Inclua o nome do usuário e a senha no corpo da solicitação JSON, no formato a seguir:

   ```
   {
       "username" : name,
       "password" : password
   }
   ```

   b) Armazene o token LTPA que é retornado da solicitação no armazenamento de cookie local. Por padrão, esse token LTPA tem um prefixo `LtpaToken2`.
2. Autentique as solicitações REST com o token LTPA armazenado como um cookie com cada solicitação.
   Para solicitações que usam os métodos de HTTP PUT, PATCH ou DELETE, inclua um cabeçalho `ibm-mq-rest-csrf-token`. O valor desse cabeçalho pode ser qualquer coisa, inclusive em branco.
3. Efetue logout de um usuário:
   a) Use o método HTTP DELETE no recurso `login`:

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

Deve-se fornecer o token LTPA como um cookie para autenticar a solicitação e incluir um cabeçalho `ibm-mq-rest-csrf-token`. O valor desse cabeçalho pode ser qualquer coisa, inclusive em branco

b) Processe a instrução para excluir o token LTPA do armazenamento de cookie local.

Nota: Se a instrução não for processada e o token LTPA permanecer no armazenamento de cookie local, o token LTPA poderá ser usado para autenticar solicitações REST futuras. Ou seja, quando o usuário tenta autenticar com o token LTPA após a sessão ser encerrada, uma nova sessão é criada usando o token existente.

Exemplo

O exemplo de cURL a seguir mostra como criar uma nova fila Q1, no gerenciador de filas QM1, com autenticação baseada em token em sistemas Windows:

- Efetue login e inclua o token LTPA com o prefixo `LtpaToken2` no armazenamento de cookie local. As informações sobre nome do usuário e senha são incluídas no corpo JSON. A sinalização `-c` especifica o local do arquivo para armazenar o token em:

```
curl -k https://localhost:9443/ibmmq/rest/v1/login -X POST
-H "Content-Type: application/json" --data
"{\"username\":\"mqadmin\",\"password\":\"mqadmin\"}"
-c c:\cookiejar.txt
```

- Crie uma fila. Use o método de HTTP POST com o recurso de fila, autenticando com o token LTPA. O token LTPA com o prefixo `LtpaToken2` é recuperado do arquivo `cookiejar.txt` usando a sinalização `-b`. A proteção de CSRF é fornecida pela presença do cabeçalho de HTTP `ibm-mq-rest-csrf-token`:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X POST -b
c:\cookiejar.txt -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json"
--data "{\"name\":\"Q1\"}"
```

- Efetue logout e exclua o token LTPA do armazenamento de cookie local. O token LTPA é recuperado do arquivo `cookiejar.txt` usando a sinalização `-b`. A proteção de CSRF é fornecida pela presença do cabeçalho de HTTP `ibm-mq-rest-csrf-token`. O local do arquivo `cookiejar.txt` é especificado pela sinalização `-c` para que o token LTPA seja excluído do arquivo:

```
curl -k https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1/queue -X DELETE
-H "ibm-mq-rest-csrf-token: value" -b c:\cookiejar.txt
-c c:\cookiejar.txt
```

Referências relacionadas

POST /login
GET /login
Excluir /login

Integrando o IBM MQ Console a um IFrame

O elemento HTML < iframe> pode ser usado para integrar uma página da web em outra usando um Quadro Sequencial (IFrame). Por razões de segurança, o IBM MQ Console não pode ser integrado em um IFrame por padrão. No entanto, é possível ativar um IFrame usando a propriedade de configuração **mqConsoleFrameAncestors** no servidor mqweb.

Sobre esta tarefa

O servidor mqweb mantém uma lista de permissões de origens de páginas da web que podem integrar o IBM MQ Console usando um IFrame. Uma origem é uma combinação de um esquema de URL, domínio e porta, por exemplo, `https://example.com:1234`.

É possível usar a propriedade de configuração **mqConsoleFrameAncestors** no servidor mqweb para especificar as entradas na lista.

Por padrão, **mqConsoleFrameAncestors** está em branco, o que significa que o IBM MQ Console não pode ser integrado em um IFrame.

Procedimento

Especifique uma lista de origens de páginas da web, que podem integrar o IBM MQ Console em um IFrame, inserindo o comando a seguir:

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

em que *allowedOrigins* é uma lista de origens separada por vírgulas. Cada origem deve consistir em:

- Um nome do host ou endereço IP
- Um esquema de URL opcional
- Um número de porta opcional

Observe que o nome do host pode começar com o caractere curinga (*) e o número da porta também pode usar o caractere curinga curinga (*).

As origens de exemplo são:

```
https://example.com:1234
```

que permite que qualquer página da web seja entregue a partir de `https://example.com:1234` para integrar o IBM MQ Console em um IFrame.

```
https://*.example.com:*
```

que permite qualquer página da web HTTPS com um nome do host que termina com `example.com`, e usando qualquer porta, para integrar o IBM MQ Console em um IFrame.

Exemplo

O exemplo a seguir permite que o IBM MQ Console seja integrado a um IFrame a partir de páginas da web servidas a partir de `https://site2.example.com:1234` ou `https://site2.example.com:1235`:

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

Configurando o CORS para a REST API

Por padrão, um navegador da web não permite scripts, como JavaScript, para chamar a REST API quando o script não é da mesma origem que a REST API. Ou seja, as solicitações de origem cruzada não estão ativadas. É possível configurar o Cross Origin Resource Sharing (CORS) para permitir solicitações de origem cruzada de origens especificadas.

Sobre esta tarefa

É possível acessar a REST API por meio de um navegador da web, por exemplo, por um script. Como essas solicitações são de uma origem diferente para a REST API, o navegador da web recusa a solicitação por se tratar de uma solicitação de origem cruzada. A origem será diferente se o domínio, porta ou esquema não for o mesmo.

Por exemplo, se você tiver um script hospedado em `http://localhost:1999/`, fará uma solicitação de origem cruzada se você emitir um HTTP GET em um website hospedado em `https://localhost:9443/`. Essa solicitação é uma solicitação de origem cruzada porque os números de porta e esquema (HTTP) são diferentes.

É possível ativar solicitações de origem cruzada configurando o CORS e especificando as origens que têm permissão para acessar a REST API.

Para obter mais informações sobre o CORS, consulte https://www.w3.org/TR/cors/ e https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

Procedimento

1. Visualize a configuração atual inserindo o comando a seguir:

   ```
   dspmqweb properties -a
   ```

   A entrada `mqRestCorsAllowedOrigins` especifica as origens permitidas. A entrada `mqRestCorsMaxAgeInSeconds` especifica o tempo, em segundos, que o navegador da web pode armazenar em cache os resultados de quaisquer verificações de simulação do CORS.
2. Especifique as origens que têm permissão para acessar a REST API inserindo o comando a seguir:

   ```
   setmqweb properties -k mqRestCorsAllowedOrigins -v allowedOrigins
   ```

   em que *allowedOrigins* especifica a origem da qual você deseja permitir solicitações de origem cruzada. É possível usar um asterisco entre aspas duplas, "*", para permitir todas as solicitações de origem cruzada. É possível inserir mais de uma origem em uma lista separada por vírgula, circundada por aspas duplas. Para permitir solicitações que não sejam de origem cruzada, insira aspas vazias como o valor para *allowedOrigins*.
3. Especifique o tempo, em segundos, que você deseja permitir que um navegador da web armazene em cache os resultados de quaisquer verificações prévias do CORS, inserindo o comando a seguir.

   ```
   setmqweb properties -k mqRestCorsMaxAgeInSeconds -v time
   ```

Exemplo

O exemplo a seguir mostra as solicitações de origem cruzada ativadas para `http://localhost:9883`, `https://localhost:1999` e `https://localhost:9663`. A idade máxima dos resultados em cache de quaisquer verificações prévias do CORS é configurada como 90 segundos:

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://
localhost:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

Configurando a validação do cabeçalho do host para o IBM MQ Console e a REST API

É possível configurar o servidor mqweb para restringir o acesso ao IBM MQ Console e à REST API de modo que somente as solicitações enviadas com um cabeçalho do host que corresponda a uma lista de permissões especificada sejam processadas. Um erro será retornado se um valor de cabeçalho do host que não esteja na lista de permissões for usado.

Sobre esta tarefa

O servidor mqweb usa hosts virtuais para definir a lista de permissões de cabeçalhos do host aceitáveis. Para obter mais informações sobre os hosts virtuais, consulte a documentação do WebSphere Liberty: https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html

Para completar esta tarefa, você deve ser um usuário com privilégios suficientes para editar o arquivo `mqwebuser.xml`:

- z/OS No z/OS, você deve ter acesso de gravação ao arquivo `mqwebuser.xml`.
- Multi Em todos os outros sistemas operacionais, deve-se ser um usuário privilegiado.
- Linux V 9.4.0 Se o servidor mqweb fizer parte de uma instalação independente do IBM MQ Web Server , deve-se ter acesso de gravação ao arquivo `mqwebuser.xml` no diretório de dados IBM MQ Web Server

Procedimento

1. Abra o arquivo `mqwebuser.xml`. Esse arquivo está em um dos locais a seguir:
   - Em uma instalação do IBM MQ :
     - Linux AIX Em AIX ou Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`
     - Windows Em Windows: *`MQ_DATA_PATH`*`\web\installations\`*`installationName`*`\servers\mqweb`, em que *`MQ_DATA_PATH`* é o caminho de dados IBM MQ . Esse caminho é o caminho de dados selecionado durante a instalação do IBM MQ. Por padrão, esse caminho é `C:\ProgramData\IBM\MQ`.
     - z/OS No z/OS: *`WLP_user_directory`*`/servers/mqweb`

       Em que *WLP_user_directory* é o diretório especificado quando o comando **crtmqweb** foi executado para criar a definição do servidor mqweb.
   - Linux V 9.4.0 Em uma instalação IBM MQ Web Server independente: *`MQ_OVERRIDE_DATA_PATH`*`/web/installations/MQWEBINST/servers/mqweb`

     em que *MQ_OVERRIDE_DATA_PATH* é o diretório de dados IBM MQ Web Server para o qual a variável de ambiente do **MQ_OVERRIDE_DATA_PATH** aponta.
2. Inclua ou descomente o código a seguir no arquivo `mqwebuser.xml`:

```
<virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
        <hostAlias>localhost:9080</hostAlias>
</virtualHost>
```

3. Edite o campo **<hostAlias>**, inserindo a combinação de nome de host e porta que você deseja permitir.

   Essa combinação pode ser o nome do host e o nome da porta que você usou na configuração do servidor mqweb. Por exemplo, se você usar a configuração padrão do `localhost:9443`, você pode querer usar `localhost:9443` no campo **<hostAlias>**.

   Se necessário, é possível incluir vários campos **<hostAlias>** dentro das tags **<virtualHost>** para permitir mais combinações de nome de host e porta. Por exemplo, para permitir cabeçalhos do host que usam uma porta HTTP, bem como cabeçalhos do host que usam a porta HTTPS.

Auditoria

Os registros de auditoria de operações que são executadas no IBM MQ Console e REST API podem ser produzidos ativando o comando do gerenciador de filas e eventos de configuração e no AIX, Linux, and Windows mudanças de estado significativas são registradas nos arquivos de log do servidor mqweb.

Mudanças de Estado Significativas

ALW

No AIX, Linux, and Windows, o IBM MQ Console registra mudanças de estado significativas, como mensagens nos logs do servidor mqweb. Cada mensagem indica o nome do principal autenticado que solicitou a operação.

Mudanças de estado significativas, como quando os gerenciadores de filas são criados, iniciados, terminados ou excluídos, são registradas nos arquivos `messages.log` e `console.log` do servidor mqweb no nível de criação de log [AUDIT]. Cada entrada de log indica o nome do principal autenticado que solicitou a operação.

Os arquivos `messages.log` e `console.log` podem ser localizados no local a seguir:

- Em uma instalação do IBM MQ :

- Linux AIX Em AIX ou Linux: `/var/mqm/web/installations/installationName/servers/mqweb/logs`
- Windows Em Windows: *MQ_DATA_PATH*`\web\installations\`*installationName*`\servers\mqweb\logs`, em que *MQ_DATA_PATH* é o caminho de dados IBM MQ . Esse caminho é o caminho de dados selecionado durante a instalação do IBM MQ. Por padrão, esse caminho é `C:\ProgramData\IBM\MQ`.

- Linux V 9.4.0 Em uma instalação IBM MQ Web Server independente: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb/logs`

  em que *MQ_OVERRIDE_DATA_PATH* é o diretório de dados IBM MQ Web Server para o qual a variável de ambiente do **MQ_OVERRIDE_DATA_PATH** aponta.

Para obter mais informações sobre como configurar os níveis de criação de log do servidor mqweb, consulte Configurando a criação de log.

Eventos de comando e configuração

É possível, opcionalmente, ativar eventos de comando e configuração no gerenciador de filas para fornecer informações sobre a maioria das atividades do IBM MQ Console e do REST API. Por exemplo, a criação de canais e a consulta de filas geram eventos de comando e de configuração. Para obter mais informações sobre como ativar eventos de comando e configuração, veja Controlando os eventos de configuração, de comando e de criador de logs.

Para essas mensagens de evento de comando e configuração, o campo **MQIACF_EVENT_ORIGIN** é configurado como `MQEVO_REST` e o campo **MQCACF_EVENT_APPL_IDENTITY** relata os primeiros 32 caracteres do nome principal autenticado. Se um usuário tiver a função `MQWebAdmin` ou `MQWebAdminRO` , o campo **MQCACF_EVENT_USER_ID** relatará o ID do usuário do servidor mqweb, não o nome do usuário do principal que emitiu o comando.. No entanto, se o usuário tiver a função `MQWebUser` , o **MQCACF_EVENT_USER_ID** relatará o nome do usuário do principal que emitiu o comando

Conceitos relacionados

“Auditoria” na página 485
É possível procurar por intrusões de segurança ou tentativas de intrusão usando mensagens do evento. Também é possível verificar a segurança do seu sistema usando o IBM MQ Explorer.

z/OS Considerações de segurança para o IBM MQ Console e para a REST API em z/OS

O IBM MQ Console e a REST API possuem recursos de segurança que controlam se um usuário pode emitir, exibir ou alterar comandos. Os comandos são então passados para o gerenciador de filas e o gerenciador de filas é então usado para controlar se o usuário tem permissão para emitir o comando para esse gerenciador de filas específico.

Procedimento

1. Assegure-se de que o ID do usuário da tarefa iniciada do servidor mqweb tenha autoridades apropriadas para emitir determinados comandos PCF e acessar determinadas filas. Para obter informações adicionais, consulte “Authority required by the mqweb server started task user ID” na página 545.
2. Assegure-se de que quaisquer usuários para os quais a função `MQWebUser` foi concedida tenham autoridades apropriadas.

   Os usuários do IBM MQ Console e da REST API aos quais é designada a função `MQWebUser` operam sob o contexto de segurança do principal. Esses IDs de usuário podem executar somente operações que o ID do usuário possa executar no gerenciador de filas e precisam receber acesso às mesmas filas do sistema que o espaço de endereço do servidor mqweb.

O ID do usuário da tarefa iniciada do servidor mqweb deve ter o acesso de usuário alternativo concedido para todos os usuários designados à função `MQWebUser`.

Para obter mais informações sobre como conceder autoridades apropriadas para usuários com a função `MQWebUser`, veja "Acesso aos recursos do IBM MQ requeridos para usar o IBM MQ Console ou a REST API" na página 545.

3. Opcional: Configure o TLS para o IBM MQ Console e a REST API. Para obter informações adicionais, consulte "Configuring TLS for the REST API and IBM MQ Console on z/OS" na página 546.

z/OS Authority required by the mqweb server started task user ID

On z/OS, the mqweb server started task user ID requires certain authorities to issue PCF commands and access system resources.

The mqweb server started task user ID needs:

- A z/OS UNIX user identifier (UID) to be able to use z/OS UNIX System Services.
- Access to the `hlq.SCSQAUTH` and `hlq.SCSQANL*` data sets in the IBM MQ installation.
- Read access to the IBM MQ installation files in z/OS UNIX System Services.
- Read and write access to the Liberty user directory created by the **crtmqweb** script.
- Authority to connect to the queue manager. Grant the mqweb server started task user ID *READ* access to the `hlq.BATCH` profile in the MQCONN class.
- Authority to issue IBM MQ commands and access certain queues. These details are described in "IBM MQ Console - required command security profiles" on page 234, "System queue security" on page 213, and "Profiles for context security" on page 223.
- Authority to subscribe to the `SYSTEM.FTE` topic, in order to use the REST API for MFT. Grant the mqweb server started task user ID *ALTER* access to the `hlq.SUBSCRIBE.SYSTEM.FTE` profile in the MXTOPIC class.
- If you are are configuring a SAF registry, access to various security profiles. See "Configuring a SAF registry for the IBM MQ Console and REST API" on page 528 for more information.

Connection authentication

If your queue manager has been configured to require that all batch applications provide a valid user ID and password, by setting CHKLOCL(REQUIRED), you must give the mqweb server started task user ID *UPDATE* access to the `hlq.BATCH` profile in the MQCONN class.

This authority causes connection authentication to operate in CHKLOCL(OPTIONAL) mode for the mqweb server started task user ID.

If you have not configured the queue manager to require that all batch applications provide a valid user ID and password, it is sufficient to give the user ID that starts the mqweb server task *READ* access to the `hlq.BATCH` profile in the MQCONN class.

For more information about CHCKLOCL, see "Using CHCKLOCL on locally bound applications" on page 204.

z/OS Acesso aos recursos do IBM MQ requeridos para usar o IBM MQ Console ou a REST API

As operações executadas no IBM MQ Console ou na REST API por um usuário na função `MQWebUser` ocorrem sob o contexto de segurança do usuário.

Sobre esta tarefa

Consulte "Funções no IBM MQ Console e na REST API" na página 530 para obter mais informações sobre as funções no IBM MQ Console e na REST API.

Use o procedimento a seguir para conceder a um usuário, na função MQWebUser, o acesso aos recursos do gerenciador de filas necessários para usar o IBM MQ Console ou a REST API.

Procedimento

1. Conceda acesso de usuário alternativo ao ID do usuário `mqweb server started task` a cada ID do usuário na função MQWebUser.

   Faça isso em cada gerenciador de filas que os usuários administrarão por meio do IBM MQ Console ou da REST API.

   É possível usar os comandos RACF de amostra a seguir para conceder ao usuário alternativo do ID do usuário do `mqweb server started task` acesso a um usuário na função MQWebUser :

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   em que:

   hlq
   : É o prefixo do perfil, que pode ser o nome do gerenciador de filas ou o nome do grupo de filas compartilhadas

   userId
   : É o usuário na função MQWebUser

   mqwebUserId
   : É o ID do usuário `mqweb server started task`

   Nota: Se você estiver usando segurança composta por letras maiúsculas e minúsculas, use a classe MXADMIN em vez da classe MQADMIN.
2. Conceda a cada usuário na função MQWebUser o acesso a filas do sistema que são necessárias para usar o IBM MQ Console e a REST API.

   Para fazer isso, para SYSTEM.ADMIN.COMMAND.QUEUE e SYSTEM.REST.REPLY.QUEUE, dê a cada usuário acesso UPDATE às classes MQQUEUE ou MXQUEUE, dependendo se a segurança composta por letras maiúsculas e minúsculas está ou não em uso.

   É necessário fazer isso em cada gerenciador de filas que o usuário administrará por meio do REST API, incluindo gerenciadores de filas remotas administrados por meio do gateway doadministrative REST API.
3. Para permitir que um usuário na função MQWebUser administre gerenciadores de filas remotas, conceda ao usuário acesso UPDATE ao perfil na classe MQQUEUE ou MXQUEUE, protegendo a fila de transmissão usada para enviar comandos ao gerenciador de filas remotas. Observe que é necessário fornecer ao usuário acesso UPDATE no gerenciador de filas de gateway.

   No gerenciador de filas remotas, conceda o acesso do mesmo usuário, para colocar na fila de transmissão usada para enviar mensagens de resposta do comando de volta para o gerenciador de filas de gateway.
4. Conceda aos usuários na função MQWebUser o acesso a quaisquer outros recursos necessários para executar as operações suportadas pelo IBM MQ Console e pela REST API.

   O acesso necessário para:

   - Executar operações no REST API, é descrito nas seções *Requisitos de segurança* dos recursosREST API individuais
   - Emitir comandos pelo IBM MQ Console, é descrito em “IBM MQ Console - required command security profiles” na página 234

z/OS Configuring TLS for the REST API and IBM MQ Console on z/OS

On z/OS, you can configure the mqweb server to use a RACF key ring to store certificates for secure connections with TLS, and client certificate authentication.

Before you begin

You must be a user that has write access to the `mqwebuser.xml` file, and authority to work with SAF key rings, to complete this procedure.

About this task

The default mqweb server configuration uses Java keystores for the server and trusted certificates. On z/OS, you can configure the mqweb server to use a RACF key ring, instead of the Java keystores. The server can also be configured to allow users to authenticate using a client certificate.

See Liberty: Keystores for information on using RACF key rings in Liberty.

Follow this procedure to configure the mqweb server to use a RACF key ring, and optionally configure client certificate authentication. This procedure describes the steps necessary to create and use certificates signed with your own certificate authority (CA) certificates. For production, you might prefer to use certificates obtained from an external certificate authority.

Procedure

1. Create a certificate authority (CA) certificate, which will be used to sign the server certificate. For example, enter the following RACF command:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb Certification Authority') -
    O('IBM') -
    OU('MQ')) -
  SIZE(2048) -
  WITHLABEL('mqwebCertauth')
```

2. Create a server certificate, signed with the CA certificate created in step 1, by entering the following command:

```
RACDCERT ID(mqwebUserId) GENCERT -
  SUBJECTSDN(CN('hostname') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebCertauth')) -
  WITHLABEL('mqwebServerCert')
```

   where *mqwebUserId* is the mqweb server started task user ID, and *hostname* is the host name of the mqweb server.

3. Connect the CA certificate and server certificate to a SAF key ring by entering the following commands:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebCertauth') CERTAUTH)
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebServerCert'))
```

   where *mqwebUserId* is the mqweb server started task user ID, and *keyring* is the name of the key ring you want to use.

4. Export the CA certificate to a CER file by entering the following command:

```
RACDCERT CERTAUTH EXPORT(LABEL('mqwebCertauth')) -
        DSN('hlq.CERT.MQWEBCA') -
        FORMAT(CERTDER) -
        PASSWORD('password')
```

5. FTP the exported CA certificate in binary to your workstation, and import it into your browser as a certificate authority certificate.
6. Optional: If you want to configure client certificate authentication, create and export a client certificate.
   a) Create a certificate authority (CA) certificate, which will be used to sign the client certificate. For example, enter the following RACF command:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb User CA') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
  WITHLABEL('mqwebUserCertauth')
```

b) Connect the CA certificate to a SAF key ring by entering the following command:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebUserCertauth') CERTAUTH)
```

where *mqwebUserId* is the mqweb server started task user ID, and *keyring* is the name of the key ring you want to use.

c) Create a client certificate, signed with the CA certificate. For example, enter the following command:

```
RACDCERT ID(clientUserId) GENCERT -
  SUBJECTSDN(CN('clientUserId') -
     O('IBM') -
     OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebUserCertauth')) -
  WITHLABEL('userCertLabel')
```

where *clientUserId* is the user name.

The method used to map a certificate to a principal depends on the type of user registry configured:

- If you are using a basic registry, the Common Name field in the certificate is matched against the user in the registry.
- If you are using a SAF registry, and the certificate is in the RACF database, the certificate owner, specified with the **ID** parameter when creating the certificate, is used.
- If you are using an LDAP registry, the full distinguished name in the certificate is matched against the LDAP registry.

d) Export the client certificate to a PKCS #12 file by entering the following command:

```
RACDCERT ID(mqwebUserId) EXPORT(LABEL('userCertLabel')) -
         PASSWORD('password') DSN('hlq.USER.CERT')
```

e) FTP the exported certificate in binary to your workstation. To use the client certificate with the IBM MQ Console, import it into the web browser used to access the IBM MQ Console as a personal certificate.

7. Edit the file *WLP_user_directory*/servers/mqweb/mqwebuser.xml, where *WLP_user_directory* is the directory that was specified when the **crtmqweb** script ran to create the mqweb server definition.

   Make the following changes to configure the mqweb server to use a RACF key ring:

   a) Remove, or comment out, the following line:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

   b) Add the following statements:

```
<keyStore id="defaultKeyStore" filebased="false"
          location="safkeyring://mqwebUserId/keyring"
          password="password" readOnly="true" type="JCERACFKS" />
<ssl id="thisSSLConfig" keyStoreRef="defaultKeyStore" sslProtocol="TLSv1.2"
          serverKeyAlias="mqwebServerCert" clientAuthenticationSupported="true" />
<sslDefault sslRef="thisSSLConfig"/>
```

   where:

   - *mqwebUserId* is the mqweb server started task user ID.
   - *keyring* is the name of the RACF key ring.

- *mqwebServerCert* is the label of the mqweb server certificate.

Notes: The value of **keyStore password** is ignored.

8. Restart the mqweb server by stopping and restarting the mqweb server started task.
9. Optional: Use the client certificate to authenticate:
   - To use the client certificate with the IBM MQ Console, enter the URL for the IBM MQ Console in the web browser where you installed the client certificate.
   - To use the client certificate with the REST API, provide the client certificate with each REST request.

   Notes:

   a. If you are using only certificates to authenticate to the IBM MQ Console, the browser might display a list of certificates for you to select from.

   b. If you want to use a different certificate you might need to close and restart your browser.

   c. If you are using client certificates that are not in the RACF database, you can use RACF certificate name filtering, to map certificate attributes to a user ID. For example:

   ```
   RACDCERT ID(DEPT3USR) MAP SDNFILTER(OU=DEPT1.C=US)
   ```

   maps certificates with a subject distinguished name containing OU=DEPT1 and C=US to user ID DEPT3USR.

Results

You have set up a TLS interface for the IBM MQ Console and REST API.

ALW Gerenciando chaves e certificados no AIX, Linux, and Windows

No AIX, Linux, and Windows, use os comandos **runmqakm** e V 9.4.0 V 9.4.0 **runmqktool** para gerenciar chaves, certificados e solicitações de certificado

Sobre esta tarefa

O comando **runmqakm** fornece funções semelhantes às do **gskitcapicmd**. V 9.4.0 V 9.4.0 O comando **runmqktool** fornece funções semelhantes às do utilitário de gerenciamento de certificado do Java **keytool** . Antes de usar os comandos **runmqakm** ou **runmqktool** , certifique-se de que as variáveis de ambiente de sistemas estejam configuradas corretamente executando o comando **setmqenv**

O comando **runmqktool** requer que o componente JRE do IBM MQ seja instalado Se esse componente não for instalado, será possível usar o comando **runmqakm** no lugar.

Se for necessário gerenciar certificados de TLS de um modo que seja compatível com o FIPS, você deverá usar o comando **runmqakm**. Isso ocorre porque o comando **runmqakm** suporta criptografia mais avançada.

Procedimento

- Use os comandos **runmqakm** e **runmqktool** para concluir as seguintes ações:
  - Crie um repositório de chaves CMS e PKCS #12 suportado pelo IBM MQ .
  - Criar solicitações de certificado
  - Exportar certificados.
  - Importe certificados pessoais e certificados de CA.
  - Gerenciar certificados autoassinados.
  - Criar, extrair e incluir chaves secretas.

Informações relacionadas

Keytool

ALW Comandos runmqakm e runmqktool em AIX, Linux, and Windows

Em sistemas AIX, Linux, and Windows , use os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool) para gerenciar chaves e certificados..

Nota: V 9.4.0 V 9.4.0

Em IBM MQ 9.4.0, os comandos **runmqckm** e **strmqikm** são removidos. O comando **runmqktool** pode ser usado em vez do comando **runmqckm** para gerenciar os repositórios de chaves PKCS #12 e JKS Não há nenhuma substituição para a GUI do **strmqikm**

Os comandos **runmqckm** e **runmqktool** têm as seguintes diferenças importantes:

- O comando **runmqktool** não suporta arquivos stash para armazenar senhas do repositório de chave.. A senha para acessar um repositório de chaves deve sempre ser fornecida para o comando **runmqktool** quando ele for executado, como um parâmetro para o comando ou em resposta a um prompt emitido pelo comando
- O comando **runmqktool** não suporta repositórios de chaves do CMS Portanto, para exportar um certificado de um JKS para um repositório de chaves CMS , deve-se concluir as etapas a seguir:
  1. Use o comando **runmqktool -importkeystore** para copiar o certificado do repositório de chave do JKS para um repositório de chaves PKCS #12 intermediário Para obter mais informações sobre a exportação de um certificado, consulte “Exportando um certificado pessoal de um repositório de chaves no AIX, Linux, and Windows” na página 560
  2. Use o comando **runmqakm -cert -import** para importar o certificado do repositório de chaves PKCS #12 intermediário no repositório de chaves CMS . Para obter mais informações sobre a importação de um certificado, consulte “Importando um certificado pessoal em um repositório de chaves no AIX, Linux, and Windows” na página 562

Os comandos IBM MQ a seguir podem ser usados para gerenciar chaves e certificados:

runmqakm

- Fornece funções semelhantes às do **gskitcapicmd**.
- Suporta repositórios de chaves CMS e PKCS #12 .
- Suporta a criação de um arquivo stash para armazenar a senha do repositório de chaves criptografada
- Certificado como compatível com FIPS 140-2 e pode ser configurado para operar de maneira compatível com FIPS com o parâmetro **-fips** .

V 9.4.0 V 9.4.0 **runmqktool**

- Fornece funções que são semelhantes às do comando Java **keytool** .
- Suporta repositórios de chaves PKCS #12, JKS e JCE.
- Requer que o componente IBM MQ Java runtime environment (JRE) esteja instalado.

Se você precisar gerenciar certificados de uma maneira compatível com FIPS, use o comando **runmqakm** .

Para obter mais informações sobre o comando **runmqakm** , consulte runmqakm.

V 9.4.0 V 9.4.0 Para obter mais informações sobre o comando **runmqktool** , consulte runmqktool..

Os tópicos nesta seção contêm exemplos de como esses comandos são usados para concluir as tarefas comuns de gerenciamento de certificado

ALW Criando um certificado pessoal autoassinado no AIX, Linux, and Windows

Siga este procedimento para criar um certificado pessoal auto-assinado em um repositório de chaves.

Nota: O IBM MQ não suporta algoritmos SHA-3 ou SHA-5. É possível usar os nomes de algoritmos de assinatura digital SHA384WithRSA e SHA512WithRSA porque ambos os algoritmos são membros da família do SHA-2.

Deprecated Os nomes dos algoritmos de assinatura digital SHA3WithRSA e SHA5WithRSA são descontinuados porque eles são uma forma abreviada do SHA384WithRSA e do SHA512WithRSA, respectivamente.

É possível criar um certificado autoassinado usando o comando **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Para obter mais informações sobre o motivo pelo qual você pode desejar usar certificados autoassinados, consulte Usando certificados autoassinados para autenticação mútua de dois gerenciadores de filas.

Nem todos os certificados digitais podem ser usados com todos os CipherSpecs. Assegure-se de criar um certificado compatível com os CipherSpecs que você usa. O IBM MQ suporta três tipos diferentes de CipherSpec. Para obter mais informações, consulte “Interoperabilidade da curva elíptica e do RSA de CipherSpecs” na página 49.

Para usar os CipherSpecs do Tipo 1 (aqueles com nomes que começam com ECDHE_ECDSA_), deve-se usar o comando **runmqakm** para criar o certificado e deve-se especificar um parâmetro de algoritmo de assinatura Elliptic Curve ECDSA. Por exemplo, ao especificar o parâmetro **-sig_alg** EC_ecdsa_with_SHA384,

Usando o runmqakm

Emita o seguinte comando para criar um certificado pessoal autoassinado com o comando **runmqakm** :

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

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves O repositório de chaves já deve existir..

-pw *password*
Especifica a senha para o repositório de chaves

-label *label*
Especifica o rótulo certificado. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

O rótulo de um certificado TLS que é usado pelo IBM MQ é o valor do atributo **CERTLABL** se ele for configurado ou o padrão ibmwebspheremq com o nome do gerenciador de filas ou o ID do usuário IBM MQ MQI client anexado, tudo em minúsculas. Para obter mais informações, consulte “Rótulos de Certificados Digitais, Entendendo os Requisitos” na página 27.

-dn *distinguished_name*
Especifica o nome distinto X.500 colocado entre aspas duplas. Pelo menos um atributo é necessário no nome distinto É possível fornecer diversos atributos OU e DC.

Nota: O comando **runmqakm** se refere ao atributo de código de endereçamento postal como POSTALCODE, não PC. Sempre especifique POSTALCODE no parâmetro **-dn** ao usar o comando **runmqakm** para solicitar certificados com um código postal.

-size *key_size*
Especifica o tamanho da chave. O valor pode ser 512, 1024ou 2048.

-x509version *versão*
A versão do certificado X.509 a ser criado. O valor pode ser de 1, 2 ou 3. O padrão é 3.

-expire *days*
O prazo de expiração em dias do certificado. O padrão é 365 dias para um certificado.

-fips
Especifica que o comando é executado no modo FIPS. Somente o componente FIPS IBM Crypto for C (ICC) é usado e esse componente deve ser inicializado com êxito no modo FIPS Quando no modo FIPS, o componente ICC usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

-sig_alg
Especifica o algoritmo hash usado quando o certificado é criado. Esse algoritmo hash é usado para criar a assinatura associada ao certificado. O valor pode ser `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` ou `EC_ecdsa_with_SHA512`.

O valor padrão é `SHA1WithRSA`.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert.

Usando o runmqktool

V 9.4.0 V 9.4.0

Emita o seguinte comando para criar um certificado pessoal autoassinado com o comando **runmqktool** :

```
runmqktool -genkeypair -keystore filename -storepass password -storetype store_type
           -alias label -dname distinguished_name -validity days
           -keyalg key_algorithm -keysize key_size -sigalg signature_algorithm
```

em que:

-keystore *nome do arquivo*
Especifica o nome do repositório de chaves. O repositório de chaves será criado se ele não existir

-storepass *senha*
Especifica a senha do repositório de chave..

-storetype *store_type*
Especifica o tipo de repositório de chaves

-alias *label*
Especifica o rótulo certificado. O rótulo certificado é convertido em minúsculas.

-dname *distinguished_name*
Especifica o nome distinto X.500 para o certificado entre aspas duplas.

-validade *dias*
Especifica o número de dias durante os quais o certificado é válido

-keyalg *key_algorithm*
Especifica o algoritmo usado para criar o par de chaves.

-keysize *key_size*
Especifica o tamanho da chave.

-sigalg *signature_algorithm*
Especifica o algoritmo usado para assinar o certificado. Para obter mais informações sobre os algoritmes de assinatura que podem ser especificados, consulte Algoritmos de assinatura.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte genkeypair..

ALW Solicitando um certificado pessoal no AIX, Linux, and Windows

Siga este procedimento para criar uma solicitação para um certificado pessoal

Nota: O IBM MQ não suporta algoritmos SHA-3 ou SHA-5. É possível usar os nomes de algoritmos de assinatura digital SHA384WithRSA e SHA512WithRSA porque ambos os algoritmos são membros da família do SHA-2.

Deprecated Os nomes dos algoritmos de assinatura digital SHA3WithRSA e SHA5WithRSA são descontinuados porque eles são uma forma abreviada do SHA384WithRSA e do SHA512WithRSA, respectivamente.

É possível solicitar um certificado pessoal usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Nem todos os certificados digitais podem ser usados com todos os CipherSpecs. Assegure-se de criar um certificado compatível com os CipherSpecs que você usa. O IBM MQ suporta três tipos diferentes de CipherSpec. Para obter mais informações, consulte “Interoperabilidade da curva elíptica e do RSA de CipherSpecs” na página 49.

Para usar os CipherSpecs do Tipo 1 (aqueles com nomes que começam com ECDHE_ECDSA_), deve-se usar o comando **runmqakm** para criar o certificado e deve-se especificar um parâmetro de algoritmo de assinatura Elliptic Curve ECDSA. Por exemplo, ao especificar o parâmetro **-sig_alg** EC_ecdsa_with_SHA384,

Se você estiver usando o hardware de criptografia, consulte “Solicitando um Certificado Pessoal para o Hardware PKCS #11” na página 572.

Usando o runmqakm

Emita o comando a seguir para criar uma solicitação de certificado com o comando **runmqakm** :

```
runmqakm -certreq -create -db filename -pw password -label label
         -dn distinguished_name -size key_size
         -file filename -fips -sig_alg algorithm
```

em que:

-db *filename*
Especifica o nome completo do arquivo de um repositório de chaves O repositório de chaves já deve existir..

-pw *password*
Especifica a senha para o repositório de chaves

-label *label*
Especifica o rótulo certificado. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

O rótulo de um certificado TLS que é usado pelo IBM MQ é o valor do atributo **CERTLABL** se ele for configurado ou o padrão ibmwebspheremq com o nome do gerenciador de filas ou o ID do usuário IBM MQ MQI client anexado, tudo em minúsculas. Para obter mais informações, consulte “Rótulos de Certificados Digitais, Entendendo os Requisitos” na página 27.

-dn *distinguished_name*
Especifica o nome distinto X.500 colocado entre aspas duplas. Pelo menos um atributo é necessário no nome distinto É possível fornecer diversos atributos OU e DC.

Nota: O comando **runmqakm** se refere ao atributo de código de endereçamento postal como POSTALCODE, não PC. Sempre especifique POSTALCODE no parâmetro **-dn** ao usar o comando **runmqakm** para solicitar certificados com um código postal.

-size *key_size*
Especifica o tamanho da chave. O valor pode ser 512, 1024ou 2048.

-file *filename*
Especifica o nome do arquivo para a solicitação de certificado.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que são validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

-sig_alg
Especifica o algoritmo hash usado quando a solicitação de certificado é criada. Esse algoritmo hash é usado para criar a assinatura associada à solicitação de certificado. O valor pode ser 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 ou EC_ecdsa_with_SHA512.

O valor padrão é SHA1WithRSA.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -certreq .

Usando o runmqktool

V 9.4.0 V 9.4.0

Para poder criar uma solicitação de certificado com o comando **runmqktool** , deve-se gerar um par de chaves usando o comando **runmqktool -genkeypair** . Para obter mais informações sobre o comando **runmqktool -genkeypair** , consulte “Criando um certificado pessoal autoassinado no AIX, Linux, and Windows” na página 551..

Emita o comando a seguir para criar uma solicitação de certificado com o comando **runmqktool** :

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

em que:

-keystore *nome do arquivo*
Especifica o nome do repositório de chaves.

-storepass *senha*
Especifica a senha do repositório de chave..

-alias *label*
Especifica o rótulo certificado. Este é o rótulo de certificado que foi especificado quando o par de chaves foi gerado O rótulo do certificado não faz distinção entre maiúsculas e minúsculas

-file *filename*
Especifica o nome do arquivo para a solicitação de certificado.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte certreq.

O Que Fazer A Seguir

Enviar uma solicitação de certificado para uma CA. Quando você receber o certificado assinado da CA, inclua o certificado assinado no repositório de chaves. Para obter informações adicionais, consulte “Recebendo certificados pessoais em um repositório de chaves no AIX, Linux, and Windows” na página 555.

ALW Renovando um certificado pessoal existente no AIX, Linux, and Windows

Um certificado pessoal possui uma data de expiração, após a qual o certificado não pode mais ser usado. Siga este procedimento para renovar um certificado pessoal antes que expire.

É possível renovar um certificado pessoal usando o comando **runmqakm** (GSKCapiCmd)..

Se você tiver um requisito para usar tamanhos maiores de chaves para certificados pessoais, não é possível renovar um certificado existente. Deve-se substituir sua chave existente seguindo as etapas descritas em "Solicitando um certificado pessoal no AIX, Linux, and Windows" na página 553 para criar uma nova solicitação de certificado que usa os tamanhos chave necessários.

Usando o runmqakm

Emita o comando a seguir para criar uma solicitação de certificado para renovar um certificado pessoal com o comando **runmqakm** :

```
runmqakm -certreq -recreate -db filename -pw password
         -label label -target filename
```

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

-label *label*
Especifica o rótulo certificado. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-target *filename*
Especifica o nome do arquivo para a solicitação de certificado.

O Que Fazer A Seguir

Enviar uma solicitação de certificado para uma CA. Quando você receber o certificado assinado da CA, inclua o certificado assinado no repositório de chaves. Para obter informações adicionais, consulte "Recebendo certificados pessoais em um repositório de chaves no AIX, Linux, and Windows" na página 555.

ALW Recebendo certificados pessoais em um repositório de chaves no AIX, Linux, and Windows

Use este procedimento para receber um certificado pessoal para o repositório de chaves

Após a autoridade de certificação (CA) enviar um novo certificado pessoal, inclua-o no repositório de chaves a partir do qual você gerou a nova solicitação de certificado. Se a CA enviar o certificado como parte de uma mensagem de e-mail, copie o certificado em um arquivo separado.

Antes de incluir o certificado pessoal assinado por CA no repositório de chaves, conclua as etapas em "Incluindo um certificado de autoridade de certificação ou a parte pública de um certificado confiável em um repositório de chaves no AIX, Linux, and Windows" na página 559 para incluir o certificado de CA no repositório de chaves.

É possível receber um certificado pessoal em um repositório de chave usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool).. Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Se você estiver usando o hardware de criptografia, consulte o "Recebendo um certificado pessoal no hardware do PKCS #11" na página 573.

Usando o `runmqakm`

Emita o seguinte comando para incluir um certificado pessoal em um repositório de chaves com o comando **`runmqakm`** :

```
runmqakm -cert -receive -file filename -format format
         -db filename -pw password -fips
```

em que:

-file *filename*
Especifica o nome completo do arquivo do certificado pessoal.

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves O repositório de chaves já deve existir e deve ser o mesmo repositório no qual você criou a solicitação de certificado

-pw *password*
Especifica a senha para o repositório de chaves

-format *format*
Especifica o formato do certificado. O valor pode ser `ascii` para ASCII codificado na Base64 ou `binary` para dados DER binários. O padrão é `ascii`.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **`runmqakm`** falhará.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert .

Usando o `runmqktool`

V 9.4.0 V 9.4.0

Emita o seguinte comando para incluir um certificado pessoal em um repositório de chaves com o comando **`runmqktool`** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

em que:

-keystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves O repositório de chaves já deve existir e deve ser o mesmo repositório no qual você criou a solicitação de certificado

-storepass *senha*
Especifica a senha para o repositório de chaves

-alias *label*
Especifica o rótulo do certificado que foi utilizado para criar o pedido de certificado O rótulo certificado é convertido em minúsculas.

-file *filename*
Especifica o nome completo do arquivo do certificado pessoal.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte importcert..

O Que Fazer A Seguir

Se o certificado for incluído no repositório de chave TLS do gerenciador de filas, emita o comando do MQSC **`REFRESH SECURITY TYPE(SSL)`** para atualizar o cache do repositório de chaves TLS do gerenciador de filas

ALW Extraindo um certificado de autoridade de certificação de um repositório de chaves no AIX, Linux, and Windows

Siga este procedimento para extrair um certificado de autoridade de certificação (CA) de um repositório de chaves.

É possível extrair um certificado de autoridade de certificação de um repositório de chaves usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Usando o runmqakm

Emita o comando a seguir para extrair um certificado CA com o comando **runmqakm** :

```
runmqakm -cert -extract -db filename -pw password -label label
         -target filename -format format -fips
```

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

-label *label*
Especifica o rótulo do certificado CA. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-target *filename*
Especifica o nome completo do arquivo de destino.

-format *format*
Especifica o formato do certificado. O valor pode ser `ascii` para ASCII codificado na Base64 ou `binary` para dados DER binários. O padrão é `ascii`.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert .

Usando o runmqktool

V 9.4.0 V 9.4.0

Emita o comando a seguir para extrair um certificado CA com o comando **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

em que:

-keystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves

-storepass *senha*
Especifica a senha para o repositório de chaves

-alias *label*
Especifica o rótulo do certificado CA. O rótulo do certificado não faz distinção entre maiúsculas e minúsculas

-file *filename*
Especifica o nome completo do arquivo de destino.

-rfc
Especifica que o arquivo de saída está no formato ASCII Base64-encoded , conforme definido pelo padrão RFC 1421 da Internet Se essa opção não for especificada, o arquivo de saída estará no formato binário

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte exportcert

ALW Extraindo a parte pública de um certificado autoassinado de um repositório de chaves no AIX, Linux, and Windows

Siga este procedimento para extrair a parte pública de um certificado autoassinado de um repositório de chaves.

É possível extrair a parte pública de um certificado de um repositório de chaves usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Usando o runmqakm

Emita o comando a seguir para extrair a parte pública de um certificado autoassinado com o comando **runmqakm** :

```
runmqakm -cert -extract -db filename -pw password -label label
           -target filename -format format -fips
```

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

-label *label*
Especifica o rótulo do certificado CA. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-target *filename*
Especifica o nome completo do arquivo de destino.

-format *format*
Especifica o formato do certificado. O valor pode ser `ascii` para Base64-encoded ASCII ou `binary` para dados DER binários. O padrão é `ascii`.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert.

Usando o runmqktool

V 9.4.0 V 9.4.0

Emita o comando a seguir para extrair a parte pública de um certificado autoassinado com o comando **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

em que:

-keystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves

-storepass *senha*
Especifica a senha para o repositório de chaves

-alias *label*
Especifica o rótulo do certificado CA. O rótulo do certificado não faz distinção entre maiúsculas e minúsculas

-file *filename*
Especifica o nome completo do arquivo de destino.

-rfc
Especifica que o arquivo de saída está no formato ASCII Base64-encoded , conforme definido pelo padrão RFC 1421 da Internet Se essa opção não for especificada, o arquivo de saída estará no formato binário

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte exportcert

ALW Incluindo um certificado de autoridade de certificação ou a parte pública de um certificado confiável em um repositório de chaves no AIX, Linux, and Windows

Siga este procedimento para incluir um certificado CA ou a parte pública de um certificado confiável em um repositório de chaves.

É possível incluir um certificado de autoridade de certificação ou a parte pública de um certificado confiável em um repositório de chaves usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Se o certificado que deseja incluir estiver em uma cadeia de certificados, será necessário incluir também todos os certificados que estão acima dele na cadeia. É necessário incluir os certificados em ordem estritamente decrescente iniciando da raiz, seguido pelo certificado de CA logo abaixo dela na cadeia, e assim por diante.

Nota:

- Assegure-se de que o certificado esteja na codificação ASCII (UTF-8) ou binária (DER).
- Devido a uma restrição no comando IBM Java 8 **keytool** , o **runmqktool** não pode importar certificados no formato de codificação para impressão (também conhecido como codificação Base64) conforme definido pelo Internet RFC 1421 se o arquivo contiver comentários. Para importar um certificado no formato de codificação para impressão, remova todos os comentários do arquivo.. O arquivo deve começar com uma sequência que começa com "-----BEGIN" e terminar com uma sequência que começa com "-----END"..

Usando o runmqakm

Emita o seguinte comando para incluir um certificado confiável em um repositório de chaves com o comando **runmqakm** :

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

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves O repositório de chaves já deve existir..

-pw *password*
Especifica a senha para o repositório de chaves

-label *label*
Especifica o rótulo certificado. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-file *filename*
Especifica o nome do arquivo que contém o certificado.

-format *ascii*
Especifica o formato do certificado. O valor pode ser `ascii` para Base64-encoded ASCII ou `binary` para dados DER binários. O padrão é `ascii`.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert .

Usando o runmqktool

V 9.4.0 V 9.4.0

Emita o seguinte comando para incluir um certificado confiável em um repositório de chaves com o comando **runmqktool** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

em que:

-keystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves O repositório de chaves será criado se ele não existir

-storepass *senha*
Especifica a senha para o repositório de chaves

-alias *label*
Especifica o rótulo certificado. O rótulo certificado é convertido em minúsculas.

-file *filename*
Especifica o nome completo do arquivo do certificado pessoal.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte importcert..

ALW Exportando um certificado pessoal de um repositório de chaves no AIX, Linux, and Windows

Siga este procedimento para exportar um certificado pessoal de um repositório de chaves.

Exportar um certificado copia o certificado e suas chaves públicas e privadas associadas em outro repositório de chaves.

É possível exportar um certificado de um repositório de chaves usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Usando o runmqakm

Emita o comando a seguir para exportar um certificado com o comando **runmqakm** :

```
runmqakm -cert -export -db filename -pw password -label label
         -target filename -target_pw password -target_type type
         -encryption strength -fips
```

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves que contém o certificado

-pw *password*
Especifica a senha para o repositório de chaves que contém o certificado

-label *label*
Especifica o rótulo do certificado a ser exportado O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-target *filename*
Especifica o nome completo do arquivo do repositório de chaves de origem. O repositório de chaves será criado se ele não existir

-target_pw *senha*
Especifica a senha para o repositório de chaves de origem.

-target_type *tipo*
Especifica o tipo do repositório de chaves de origem. O valor pode ser `cms` ou `pkcs12`. O padrão é `cms`..

-encryption *força*
Especifica a intensidade da criptografia que é usada no comando de exportação de certificado O valor pode ser `strong` ou `weak`. O padrão é `strong`.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert.

Usando o runmqktool

Emita o comando a seguir para exportar um certificado com o comando **runmqktool** :

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

em que:

-srckeystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves que contém o certificado

-srcstorepass *password*
Especifica a senha para o repositório de chaves que contém o certificado

-destkeystore *nome do arquivo*
Especifica o nome completo do arquivo do repositório de chaves de origem. O repositório de chaves será criado se ele não existir

-deststorepass *senha*
Especifica a senha para o repositório de chaves de origem.

-destkeypass *senha*
Especifica a senha para proteger a chave no repositório de chave de destino Se esse parâmetro não for especificado, a chave será protegida com a senha usada para proteger a chave no repositório de chaves de origem.

-deststoretype *tipo*
Especifica o tipo do repositório de chaves de origem.

-srcalias *label*
Especifica o rótulo do certificado a ser exportado O rótulo do certificado não faz distinção entre maiúsculas e minúsculas

-destalias *label*
Especifica o rótulo do certificado no repositório de chaves de destino Se esse parâmetro não for especificado, o mesmo rótulo será designado ao certificado como no repositório de chaves de origem.

O rótulo certificado é convertido em minúsculas.

-file *filename*
Especifica o nome completo do arquivo de destino.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte importkeystore

ALW Importando um certificado pessoal em um repositório de chaves no AIX, Linux, and Windows

Siga este procedimento para importar um certificado pessoal para um repositório de chaves

Importar um certificado copia o certificado e suas chaves públicas e privadas associadas de um repositório de chaves para outro repositório de chaves.

Antes de importar um certificado pessoal para um repositório de chaves, deve-se primeiro incluir a cadeia válida completa de emissão de certificados de CA no repositório de chaves. Para obter informações adicionais, consulte “Incluindo um certificado de autoridade de certificação ou a parte pública de um certificado confiável em um repositório de chaves no AIX, Linux, and Windows” na página 559.

É possível importar um certificado para um repositório de chaves usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Usando o runmqakm

Emita o comando a seguir para importar um certificado com o comando **runmqakm** :

```
runmqakm -cert -import -file filename -pw password -type type
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips
```

em que:

-file *filename*
Especifica o nome completo do arquivo completo do repositório de chaves que contém o certificado

-pw *password*
Especifica a senha para o repositório de chaves que contém o certificado

-type *tipo*
Especifica o tipo de repositório de chaves que contém o certificado O valor pode ser cms ou pkcs12. O padrão é cms..

-target *filename*
Especifica o nome completo do arquivo do repositório de chaves de origem. O repositório de chaves será criado se ele não existir

-target_pw *senha*
Especifica a senha para o repositório de chaves de origem.

-target_type *tipo*
Especifica o tipo do repositório de chaves de origem. O valor pode ser `cms` ou `pkcs12`. O padrão é `cms`..

-label *label*
Especifica o rótulo do certificado a ser importado do repositório de chaves de origem. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-new_label *label*
Especifica o rótulo designado ao certificado no repositório de chaves de destino. Se esse parâmetro não for especificado, o mesmo rótulo será designado ao certificado como no repositório de chaves de origem.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert .

Usando o `runmqktool`

V 9.4.0 V 9.4.0

Emita o comando a seguir para importar um certificado com o comando **runmqktool** :

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

em que:

-srckeystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves que contém o certificado

-srcstorepass *password*
Especifica a senha para o repositório de chaves que contém o certificado

-destkeystore *nome do arquivo*
Especifica o nome completo do arquivo do repositório de chaves de origem. O repositório de chaves será criado se ele não existir

-deststorepass *senha*
Especifica a senha para o repositório de chaves de origem.

-destkeypass *senha*
Especifica a senha para proteger a chave no repositório de chave de destino Se esse parâmetro não for especificado, a chave será protegida com a senha usada para proteger a chave no repositório de chaves de origem.

Nota: Para um repositório de chaves PKCS #12 , a chave deve ser protegida com a mesma senha do repositório de chaves de destino.

-deststoretype *tipo*
Especifica o tipo do repositório de chaves de origem.

-srcalias *label*
Especifica o rótulo do certificado para no repositório de chaves de origem O rótulo do certificado não faz distinção entre maiúsculas e minúsculas

-destalias *label*
Especifica o rótulo do certificado no repositório de chaves de destino Se esse parâmetro não for especificado, o mesmo rótulo será designado ao certificado como no repositório de chaves de origem.

O rótulo certificado é convertido em minúsculas.

-file *filename*
Especifica o nome completo do arquivo de destino.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte importkeystore

ALW Importando um certificado pessoal a partir de um arquivo .pfx Microsoft

Siga este procedimento para importar um certificado de umMicrosoft Arquivo .pfx ativadoAIX, Linux, and Windows .

Um arquivo .pfx pode conter dois certificados relacionados à mesma chave. Um é um certificado pessoal ou do site que contém uma chave pública e privada. O outro é um certificado CA (signatário) que contém apenas uma chave pública. Esses certificados não podem coexistir no mesmo repositório de chaves do CMS , portanto, apenas um deles pode ser importado

O rótulo do certificado é anexado apenas ao certificado de assinante O certificado pessoal é identificado por um UUID (Unique User Identifier) gerado pelo sistema. Siga este procedimento para importar um certificado pessoal de um arquivo .pfx e configurar o rótulo do certificado pessoal para o rótulo designado ao certificado de autoridade de certificação no arquivo .pfx. Os certificados de CA de emissão já devem estar incluídos no banco de dados de chaves de destino.

Usando o `runmqakm`

Emita o comando a seguir para importar um certificado de um arquivo .pfx com o comando **`runmqakm`** :

```
runmqakm -cert -import -file filename -pw password -type pkcs12
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips -pfx
```

em que:

-file *filename*
Especifica o nome completo do arquivo .pfx.

-pw *password*
Especifica a senha do arquivo .pfx.

-type pkcs12
Especifica o tipo do repositório de chaves..

-target *filename*
Especifica o nome completo do arquivo do repositório de chaves de origem. O repositório de chaves será criado se ele não existir

-target_pw *senha*
Especifica a senha para o repositório de chaves de origem.

-target_type *tipo*
Especifica o tipo do repositório de chaves de origem. O valor pode ser `cms` ou `pkcs12`. O padrão é `cms`..

-label *label*
Especifica o rótulo do certificado a ser importado do repositório de chaves de origem. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-new_label *label*
Especifica o rótulo designado ao certificado no repositório de chaves de destino. Se esse parâmetro não for especificado, o mesmo rótulo será designado ao certificado como no repositório de chaves de origem.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **`runmqakm`** falhará.

-pfx
Indica que o repositório de chaves de origem usa o formato PFX.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert .

ALW Importando um certificado pessoal a partir de um arquivo PKCS #7

Siga este procedimento para importar um certificado de um arquivo PKCS #7 no AIX, Linux, and Windows.

Use o comando **runmqakm** para importar certificados de um arquivo PKCS #7 no AIX, Linux, and Windows.

Incluindo um certificado de CA ou a parte pública de um certificado confiável

Emita o comando a seguir para incluir um certificado de CA, ou a parte pública de um certificado confiável, a partir de um arquivo PKCS #7 :

```
runmqakm -cert -add -db filename -pw password -type type
         -label label -file filename
```

em que:

-db *filename*
Especifica o nome completo do repositório de chaves.

-pw *password*
Especifica a senha para o repositório de chaves

-type *tipo*
Especifica o tipo do repositório de chaves..

-label *label*
Especifica o rótulo do certificado para incluir. O rótulo do certificado faz distinção entre maiúsculas e minúsculas

O rótulo é designado ao primeiro certificado incluído. Todos os outros certificados, se estiverem presentes, serão identificados com o nome do assunto.

-file *filename*
Especifica o nome completo do arquivo PKCS #7 ..

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert.

Importando um certificado pessoal

Emita o comando a seguir para importar um certificado pessoal de um arquivo PKCS #7 :

```
runmqakm -cert -import -file filename -pw password -type pkcs7
         -target filename -target_pw password -target_type type
         -label label -new_label label
```

em que:

-file *filename*
Especifica o nome completo do arquivo PKCS #7 ..

-pw *password*
Especifica a senha do arquivo PKCS #7 .

-type pkcs7
Especifica o tipo do arquivo PKCS #7 ..

-target *filename*
Especifica o nome completo do arquivo do repositório de chaves de origem. O repositório de chaves será criado se ele não existir

-target_pw *senha*
Especifica a senha para o repositório de chaves de origem.

-target_type *tipo*
Especifica o tipo do repositório de chaves de origem. O valor pode ser `cms` ou `pkcs12`. O padrão é `cms`..

-label *label*
Especifica o rótulo do certificado a ser importado do arquivo PKCS #7 . O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-new_label *label*
Especifica o rótulo designado ao certificado no repositório de chaves de destino. Se esse parâmetro não for especificado, o mesmo rótulo será designado ao certificado como no repositório de chaves de origem.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert.

ALW Listando os certificados em um repositório de chaves no AIX, Linux, and Windows

Use este procedimento para listar os certificados que estão no repositório de chaves.

É possível exibir informações sobre os certificados que estão em um repositório de chaves usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool).

Usando o runmqakm

- Emita o comando a seguir para listar os rótulos dos certificados em um repositório de chaves com o comando **runmqakm** :

```
runmqakm -cert -list -db filename -pw password
```

- Emita o seguinte comando para listar os detalhes de um certificado em um repositório de chaves com o comando **runmqakm** :

```
runmqakm -cert -details -showOID -db filename -pw password
         -label label
```

em que:

-file *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

-label *label*
Especifica o rótulo do certificado a ser listado O rótulo do certificado faz distinção entre maiúsculas e minúsculas

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert .

Usando o runmqktool

V 9.4.0 V 9.4.0

- Emita o comando a seguir para listar os rótulos dos certificados em um repositório de chaves com o comando **runmqktool** :

```
runmqktool -list -keystore filename -storepass password
```

- Emita o seguinte comando para listar os detalhes de um certificado em um repositório de chaves com o comando **runmqktool** :

```
runmqktool -list -keystore filename -storepass password -alias label -v
```

em que:

-keystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves

-storepass *senha*
Especifica a senha para o repositório de chaves

-alias *label*
Especifica o rótulo do certificado a ser listado O rótulo do certificado não faz distinção entre maiúsculas e minúsculas

-v
Solicita a saída detalhada que inclui os detalhes do certificado

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte list

ALW Excluindo um certificado de um repositório de chaves no AIX, Linux, and Windows

Use este procedimento para excluir um certificado pessoal ou de CA de um repositório de chaves.

É possível excluir um certificado de um repositório de chaves usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool). Se precisar gerenciar certificados SSL ou TLS de uma maneira que seja compatível com FIPS, use o comando **runmqakm**.

Usando o runmqakm

Emita o seguinte comando para excluir um certificado com o comando **runmqakm** :

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

em que:

-file *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

-label *label*
Especifica o rótulo do certificado a ser excluído O rótulo do certificado faz distinção entre maiúsculas e minúsculas

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que foram validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -cert .

Usando o runmqktool

V 9.4.0 V 9.4.0

Emita o seguinte comando para excluir um certificado com o comando **runmqktool** :

```
runmqktool -delete -keystore filename -storepass password -alias label
```

em que:

-keystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves

-storepass *senha*
Especifica a senha para o repositório de chaves

-alias *label*
Especifica o rótulo do certificado a ser excluído O rótulo do certificado não faz distinção entre maiúsculas e minúsculas

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte delete..

ALW Convertendo um repositório de chaves no AIX, Linux, and Windows

Use este procedimento para converter um repositório de chaves em um tipo diferente

É possível converter uma senha do repositório de chave em um tipo diferente usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool).

Usando o runmqakm

Emita o comando a seguir para converter um repositório de chaves com o comando **runmqakm** :

```
runmqakm -keydb -convert -db filename -pw password
         -new_db filename -new_pw password
         -old_format type -new_format type
```

em que:

-file *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

-new_db *nome do arquivo*
Especifica o nome completo do arquivo do novo repositório de chaves

-new_pw *senha*
Especifica a senha para o novo repositório de chaves

-old_format *type*
Especifica o tipo atual do repositório de chaves. Os valores a seguir podem ser especificados:

- pkcs12
- cms

-new_format *type*
Especifica o novo tipo de repositório de teclas Os valores a seguir podem ser especificados:

- pkcs12
- cms

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados. Consulte runmqakm -keydb

Usando o runmqktool

V 9.4.0 V 9.4.0

Emita o comando a seguir para converter um repositório de chaves com o comando **runmqktool** :

```
runmqktool -importkeystore -srckeystore filename -destkeystore filename
```

```
            -srcstoretype type -deststoretype type
            -srcstorepass password -deststorepass password
```

em que:

-todos
Especifica que a senha também é mudada para todas as entradas protegidas com a mesma senha que o repositório de chaves.

-keystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves

-destkeystore *nome do arquivo*
Especifica o nome completo do arquivo do novo repositório de chaves

-srcstoretype *tipo*
Especifica o tipo de repositório de chaves

-deststoretype *tipo*
Especifica o novo tipo de repositório de chaves

-srcstorepass *password*
Especifica a senha para o repositório de chaves

-deststorepass *senha*
Especifica a senha para o novo repositório de chaves

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte importkeystore

ALW Mudando a senha do repositório de chaves no AIX, Linux, and Windows

Use este procedimento para alterar a senha do repositório de chaves

É possível mudar a senha do repositório de chaves usando os comandos **runmqakm** (GSKCapiCmd) ou **runmqktool** (keytool)..

Nota:

- V 9.4.0 V 9.4.0 O comando **runmqktool** permite que a senha do repositório de chave seja alterada independentemente das senhas que protegem as chaves privadas e secretas individuais Para os repositórios de chaves PKCS #12 , a senha do repositório de chaves e as senhas que protegem todas as chaves no repositório de chaves devem ser as mesmas Se o comando **runmqktool** for usado para mudar a senha do repositório de chaves, assegure-se de que o parâmetro **-all** seja especificado para que as senhas de chave também sejam mudadas
- Se a senha do repositório de chaves não estiver armazenada em um arquivo stash, você também deverá alterar a senha que está armazenada na configuração do gerenciador de filas ou em qualquer aplicativo IBM MQ client que acesse o repositório de chaves. Para obter mais informações, consulte o “Fornecendo a senha do repositório de chaves para um gerenciador de filas no AIX, Linux, and Windows” na página 305 e o “Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on AIX, Linux, and Windows” na página 307.

Usando o runmqakm

Emita o seguinte comando para alterar a senha do repositório de chaves com o comando **runmqakm** :

```
runmqakm -keydb -changepw -db filename -pw password -new_pw password -stash
```

em que:

-file *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha atual para o repositório de chaves

-new_pw *senha*
Especifica a nova senha para o repositório de chaves

-stash
Opcional. Especifique esta opção para armazenar a nova senha do repositório de chaves em um arquivo stash Não será necessário armazenar a senha em um arquivo stash se você criptografar a senha usando o sistema de proteção de senha IBM MQ no lugar.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados. Consulte runmqakm -keydb

Usando o `runmqktool`

V 9.4.0 V 9.4.0

Emita o seguinte comando para alterar a senha do repositório de chaves com o comando **`runmqktool`** :

```
runmqktool -storepasswd -all -keystore filename -storepass password
           -new password
```

em que:

-todos
Especifica que a senha também é mudada para todas as entradas protegidas com a mesma senha que o repositório de chaves.

-keystore *nome do arquivo*
Especifica o nome completo do arquivo completo do repositório de chaves

-storepass *senha*
Especifica a senha atual para o repositório de chaves

-nova *senha*
Especifica a nova senha para o repositório de chaves

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte storepasswd..

ALW Gerenciando chaves secretas emAIX, Linux, and Windows

Siga este procedimento para gerenciar chaves secretas em um repositório de chaves.

Você pode gerenciar chaves secretas usando o**`runmqakm`** (GSKCapiCmd) comando. Chaves secretas geradas usando o**`runmqktool`** O comando (keytool) não pode ser usado comIBM MQ .

Criando uma chave secreta

Emita o seguinte comando para criar uma chave secreta aleatória com o**`runmqakm`** comando:

```
runmqakm -secretkey -create -db filename -pw password
         -label label -size key_size
```

em que:

-db *filename*
Especifica o nome completo do arquivo do repositório de chaves. O repositório de chaves já deve existir.

-pw *password*
Especifica a senha do repositório de chaves.

-label *label*
Especifica o rótulo anexado à chave.

-size *key_size*
Especifica o tamanho da chave em bytes.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -secretkey .

Extraindo uma chave secreta

Emita o seguinte comando para extrair uma chave secreta com o**runmqakm** comando:

```
runmqakm -secretkey -extract -db filename -pw password
         -label label -target filename -format format
```

em que:

-db *filename*
Especifica o nome completo do arquivo do repositório de chaves. O repositório de chaves já deve existir.

-pw *password*
Especifica a senha do repositório de chaves.

-label *label*
Especifica o rótulo da chave a ser extraída.

-target *filename*
Especifica o nome completo do arquivo de destino.

-formatar *formatar*
Especifica o formato da chave no arquivo de destino. O valor pode ser`ascii` paraBase64-encoded ASCII ou`binary` para uma cópia binária da chave. O padrão é `ascii`.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -secretkey .

Adicionando uma chave secreta

Emita o seguinte comando para extrair uma chave secreta com o**runmqakm** comando:

```
runmqakm -secretkey -add -db filename -pw password
         -label label -file filename -format format
```

em que:

-db *filename*
Especifica o nome completo do arquivo do repositório de chaves. O repositório de chaves já deve existir.

-pw *password*
Especifica a senha do repositório de chaves.

-label *label*
Especifica o rótulo anexado à chave.

-file *filename*
Especifica o nome do arquivo que contém a chave.

-formatar *formatar*
Especifica o formato da chave. O valor pode ser`ascii` paraBase64-encoded ASCII ou`binary` para dados binários. O padrão é `ascii`.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -secretkey .

ALW Gerenciando certificados em hardware PKCS #11

É possível gerenciar certificados digitais no hardware de criptografia que suporta a interface PKCS #11.

Deve-se criar um repositório de chaves para preparar o ambiente IBM MQ , mesmo que você não pretenda armazenar nenhum certificado nele, mas armazenará todos os seus certificados em seu

hardware de criptografia. Um repositório de chaves é necessário para o gerenciador de filas fazer referência em seu atributo **SSLKEYR** ou para o aplicativo cliente fazer referência na variável de ambiente MQSSLKEYR. Esse repositório de chaves também será necessário se você estiver criando uma requisição de certificado

Crie o repositório de teclas usando o comando **runmqakm** (GSKCapiCmd).

Emita o seguinte comando para criar um repositório de chaves com o comando **runmqakm** :

```
runmqakm -keydb -create -db filename -pw password -type type -stash
```

em que:

-db *filename*
Especifica o nome completo do arquivo completo do repositório de chaves

-pw *password*
Especifica a senha para o repositório de chaves

-type *tipo*
Especifica o tipo de banco de dados. O valor deve ser `cms` ou `pkcs12` para um repositório de chaves usado pelo IBM MQ.

-stash
Opcional. Se especificado, a senha do repositório de chaves criptografada será salva em um arquivo

ALW *Solicitando um Certificado Pessoal para o Hardware PKCS #11*

Use este procedimento para solicitar um certificado pessoal para um gerenciador de filas ou um IBM MQ MQI client com seu hardware de criptografia.

Nota: O IBM MQ não suporta algoritmos SHA-3 ou SHA-5. É possível usar os nomes de algoritmos de assinatura digital SHA384WithRSA e SHA512WithRSA porque ambos os algoritmos são membros da família do SHA-2.

Deprecated Os nomes dos algoritmos de assinatura digital SHA3WithRSA e SHA5WithRSA são descontinuados porque eles são uma forma abreviada do SHA384WithRSA e do SHA512WithRSA, respectivamente.

Antes de criar uma solicitação de certificado em seu hardware criptográfico, conclua as etapas descritas em “Gerenciando certificados em hardware PKCS #11” na página 571 para criar um repositório de chaves.

Emita o seguinte comando para criar uma solicitação de certificado com o comando **runmqakm** (GSKCapiCmd):

```
runmqakm -certreq -create -crypto module_name -tokenlabel hardware_token
         -pw password -label label
         -dn distinguished_name -size key_size
         -file filename -fips -sig_alg algorithm
```

em que:

-crypto *module_name*
Especifica o nome completo da biblioteca do PKCS #11 fornecida com o hardware de criptografia.

-tokenlabel *hardware_token*
Especifica o rótulo do token do dispositivo criptográfico do PKCS #11.

-pw *password*
Especifica a senha para acessar o hardware criptográfico

-label *label*
Especifica o rótulo certificado.

O rótulo de um certificado TLS que é usado pelo IBM MQ é o valor do atributo **CERTLABL** se ele for configurado ou o padrão `ibmwebspheremq` com o nome do gerenciador de filas ou o ID do usuário

IBM MQ MQI client anexado, tudo em minúsculas. Para obter mais informações, consulte “Rótulos de Certificados Digitais, Entendendo os Requisitos” na página 27.

-dn *distinguished_name*
Especifica o nome distinto X.500 colocado entre aspas duplas. Pelo menos um atributo é necessário no nome distinto É possível fornecer diversos atributos OU e DC.

Nota: O comando **runmqakm** se refere ao atributo de código de endereçamento postal como POSTALCODE, não PC. Sempre especifique POSTALCODE no parâmetro **-dn** ao usar o comando **runmqakm** para solicitar certificados com um código postal.

-size *key_size*
Especifica o tamanho da chave. O valor pode ser 512, 1024ou 2048.

-file *filename*
Especifica o nome do arquivo para a solicitação de certificado.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que são validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

-sig_alg
Especifica o algoritmo hash usado quando a solicitação de certificado é criada. Esse algoritmo hash é usado para criar a assinatura associada à solicitação de certificado. O valor pode ser 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 ou EC_ecdsa_with_SHA512.

O valor padrão é SHA1WithRSA.

Para obter mais informações sobre esses parâmetros e os valores que podem ser especificados, consulte runmqakm -certreq .

O Que Fazer A Seguir

Enviar uma solicitação de certificado para uma CA. Quando você receber o certificado assinado da CA, inclua o certificado assinado no repositório de chaves. Para obter informações adicionais, consulte “Recebendo um certificado pessoal no hardware do PKCS #11” na página 573.

ALW *Recebendo um certificado pessoal no hardware do PKCS #11*

Use este procedimento para receber um certificado pessoal para um gerenciador de fila ou um IBM MQ MQI client para seu hardware de criptografia.

Inclua o certificado de CA da CA que assinou o certificado pessoal no hardware criptográfico ou no repositório de chaves secundário. Faça isso antes de receber o certificado assinado no hardware de criptografia. Para incluir um certificado de autoridade de certificação em um arquivo de repositório de chaves, siga o procedimento em “Incluindo um certificado de autoridade de certificação ou a parte pública de um certificado confiável em um repositório de chaves no AIX, Linux, and Windows” na página 559

Emita o seguinte comando para incluir um certificado pessoal em um repositório de chaves com o comando **runmqakm** (GSKCapiCmd):

```
runmqakm -cert -receive -file filename -crypto module_name
         -tokenlabel hardware_token -pw hardware_password
         -format cert_format -fips
         -secondaryDB filename -secondaryDBpw password
```

em que:

-file *filename*
Especifica o nome completo do arquivo que contém o certificado pessoal.

-crypto *module_name*
Especifica o nome completo da biblioteca do PKCS #11 fornecida com o hardware de criptografia.

-tokenlabel *hardware_token*
Especifica o rótulo do token do dispositivo criptográfico do PKCS #11.

-pw *hardware_password*
Especifica a senha para acessar o hardware criptográfico

-format *cert_format*
Especifica o formato do certificado. O valor pode ser `ascii` para ASCII codificado na Base64 ou `binary` para dados DER binários. O padrão é ASCII.

-fips
Especifica que o comando é executado no modo FIPS. Quando no modo FIPS, o componente IBM Crypto for C (ICC) usa algoritmos que são validados pelo FIPS 140-2 Se o componente ICC não inicializar no modo FIPS, o comando **runmqakm** falhará.

-secondaryDB *filename*
Especifica o nome completo do arquivo de repositório de chaves que é usado para armazenar o certificado de autoridade de certificação.

-secondaryDBpw *password*
Especifica a senha para o arquivo de repositório de chaves que é usado para armazenar o certificado de CA.

Protegendo senhas em arquivos de configuração do componente do IBM MQ

Para usar determinados recursos do IBM MQ, pode ser necessário fornecer senhas que são usadas pelo recurso As senhas que são fornecidas para o IBM MQ podem ser protegidas usando um sistema de proteção de senha

A lista a seguir explica a terminologia que é usada para cada componente que processa senhas criptografadas:

Chave inicial
A chave de criptografia usada para proteger a senha.

Chave inicial padrão
A chave de criptografia padrão usada se você não fornecer uma chave inicial quando a senha for criptografada.

Sequência de texto sem formatação
A sequência criptografada, geralmente uma senha.

Sequência de senha criptografada..
Uma sequência que contém a senha criptografada em um formato que o IBM MQ entende

Especificando a chave inicial

Para cada componente, é possível escolher especificar uma chave inicial usada para criptografar senhas.

- Se você não especificar uma chave inicial, a chave inicial padrão para o componente será usada A chave inicial padrão é a mesma para todas as instalações do IBM MQ Isso significa que uma senha criptografada com a chave inicial padrão não está protegida com segurança, pois pode ser possível para uma instalação diferente decriptografar a senha.
- Se você fornecer sua própria chave inicial exclusiva, somente os usuários com acesso à chave inicial fornecida poderão decriptografar a senha.

Atenção: Para fornecer o nível mais alto de segurança para senhas armazenadas, forneça uma chave inicial exclusiva para cada componente IBM MQ .

Se você optar por usar sua própria chave inicial, especifique uma chave inicial exclusiva para cada componente listado. A chave inicial é usada para proteger quaisquer senhas armazenadas na configuração desse componente. A mesma chave inicial também deve ser disponibilizada para o componente para que a senha seja decriptografada

A maioria dos componentes requer que a chave inicial seja fornecida em um arquivo A chave inicial que está contida no arquivo de chave inicial deve atender aos seguintes requisitos:

- Ele deve ter pelo menos um caractere.
- Ele deve ser uma única linha de texto

O comprimento máximo da chave inicial é ilimitado e quaisquer caracteres podem ser especificados. Para segurança adequada, especifique uma chave inicial que tenha pelo menos 16 caracteres. Por exemplo, seu arquivo-chave inicial pode conter a seguinte sequência:

```
Th1sIs@n3Ncypt|onK$y
```

O acesso ao arquivo de chave inicial deve ser limitado apenas aos usuários que precisam acessar a chave inicial usando as permissões de arquivo do sistema operacional

Para obter mais informações sobre os benefícios e as limitações de proteção de senha, consulte “Os limites para proteção por meio de criptografia de senha” na página 581

Protegendo senhas em cada componente do IBM MQ

Vários componentes IBM MQ podem proteger senhas armazenadas. Dependendo do componente, essas senhas podem ser fornecidas usando um dos mecanismos a seguir:

- Fornecido diretamente ao IBM MQ gerenciador de filas ou IBM MQ client.
- Especificado em uma variável de ambiente.
- Armazenado em um arquivo de configuração..

Cada componente fornece um método para criptografar senhas.. Na maioria dos componentes, as senhas devem ser criptografadas antes de serem fornecidas para o IBM MQ ou armazenadas na configuração

Importante: Uma senha criptografada que é gerada para uso com um componente não pode ser copiada no arquivo de configuração de outro componente. Uma senha criptografada para uso por um componente específico deve ser protegida com o utilitário que é fornecido pelo mesmo componente.

Os detalhes de como proteger senhas para cada componente IBM MQ que suporta a proteção de senha estão listados nas seções a seguir:

- Advanced Message Security
- “Managed File Transfer” na página 576
- “IBM MQ Internet Pass-Thru” na página 577
- “IBM MQ clients que usam hardware criptográfico” na página 578
- “Gerenciador de filas da IBM MQ” na página 579
- “IBM MQ C aplicativos clientes” na página 579
- V 9.4.0 “Configurações de HA nativa” na página 580
- V 9.4.0 “Gerenciador de filas IBM MQ (sub-rotinaAuthToken no arquivo qm.ini)” na página 581

Advanced Message Security

Os clientes Advanced Message Security (AMS) Java requerem acesso a um keystore que contém as chaves privadas que são usadas para proteger mensagens.

Advanced Message Security (AMS) Os clientes MQI ou gerenciadores de filas que são configurados para executar a intercepção de MCA podem requerer acesso ao hardware de criptografia PKCS#11 ou arquivos PEM que contêm as chaves privadas que são usadas para proteger mensagens.

Para acessar esses repositórios de chaves, uma senha deve ser fornecida no arquivo de configuração AMS chamado `keystore.conf`. Use o comando **runamscred** para proteger as informações confidenciais que estão contidas no arquivo `keystore.conf` . Por exemplo,

```
runamscred -f <keystore configuration file>
```

O comando **runamscred** protege parâmetros sensíveis dentro do arquivo especificado usando o parâmetro **-f** .

Dois comandos **runamscred** estão disponíveis em uma instalação do IBM MQ :

- Um comando MQI **runamscred** que está localizado em `<IBM MQ installation root>/bin`
- Um comando Java **runamscred** que está localizado em `<IBM MQ installation root>/java/bin`

Atenção: Para assegurar a compatibilidade,

1. Use o comando Java **runamscred** para proteger os arquivos de configuração que são usados com clientes Java AMS e o comando MQI **runamscred** para proteger arquivos de configuração para IBM MQ MQI clients que usam AMS.
2. Verifique se todas as informações confidenciais necessárias estão protegidas após a execução do comando **runamscred**
3. Forneça o arquivo que contém a senha protegida normalmente para aplicativos ativados por AMS .

Por padrão, o comando **runamscred** criptografa a senha no arquivo de configuração com a chave inicial padrão Para criptografar as senhas com uma chave inicial específica, use um dos mecanismos a seguir para especificar o nome do arquivo que contém a chave inicial, em ordem de prioridade:

1. O parâmetro **-sf** para o comando **runamscred**
2. A variável de ambiente **MQS_AMSCRED_KEYFILE** .
3. O parâmetro **amscred.keyfile** no arquivo de configuração `keystore.conf` .

Cuidado: A chave inicial padrão é a mesma para todas as instalações do IBM MQ Para proteger senhas com segurança, forneça uma chave inicial que seja exclusiva para sua instalação ao criptografar senhas.

Se você especificar um arquivo de chave inicial ao executar o comando **runamscred** para criptografar as senhas na configuração do AMS , também deverá especificar o mesmo arquivo de chave inicial quando os aplicativos AMS forem executados Os seguintes mecanismos podem ser usados para especificar o nome do arquivo-chave inicial, em ordem de prioridade:

1. A variável de ambiente **MQS_AMSCRED_KEYFILE** .
2. O parâmetro **amscred.keyfile** no arquivo de configuração `keystore.conf` .

Por padrão, o comando **runamscred** protege as credenciais com um sistema de proteção que não é compatível com AMS versões anteriores a IBM MQ 9.2 Para proteger os arquivos de configuração com o sistema de proteção de credenciais compatível com versões anteriores a IBM MQ 9.2, especifique o parâmetro **-sp 0** quando o comando **runamscred** for executado.

Managed File Transfer

Managed File Transfer (MFT) armazena credenciais que são necessárias para acessar gerenciadores de filas e outros recursos nos arquivos de propriedades XML a seguir:

MQMFTCredentials.xml
Esse arquivo contém as seguintes credenciais:

- Credenciais que são usadas para conectar ao agente, coordenação e gerenciadores de filas de comando.
- Senhas que são usadas para acessar armazenamentos de chaves que são usados para comunicações seguras

ProtocolBridgeCredentials.xml
Esse arquivo contém as credenciais que são usadas para se conectar aos Servidores de Protocolo, como FTP, SFTP e FTPS, por exemplo.

ConnectDirectCredentials.xml
Esse arquivo contém credenciais que são usadas por um agente Connect:Direct para se conectar a um nó Connect:Direct .

Para proteger informações confidenciais armazenadas nesses arquivos, use o comando fteObfuscate . Especifique o nome do arquivo que deve ser protegido usando o sinalizador **-f** Por exemplo:

```
fteObfuscate -f <File to protect>
```

Por padrão, o comando **fteObfuscate** protege as credenciais com a chave inicial padrão Para proteger as credenciais com uma chave inicial específica, use o parâmetro **-sf** para especificar o caminho para o arquivo que contém a chave inicial Por exemplo:

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

Cuidado: A chave inicial padrão é a mesma para todas as instalações do IBM MQ Para proteger senhas com segurança, forneça uma chave inicial que seja exclusiva para sua instalação ao criptografar senhas.

Atenção:

1. Verifique se todas as informações confidenciais são protegidas após a execução de **fteObfuscate**
2. Forneça o arquivo protegido normalmente para MFT.

Se você especificar um arquivo de chave inicial ao executar o comando **fteObfuscate** para proteger credenciais na configuração do MFT , também deverá especificar o mesmo arquivo de chave inicial quando MFT for iniciado. Os seguintes mecanismos podem ser usados para especificar o nome do arquivo-chave inicial, em ordem de prioridade:

1. A propriedade de sistema do **com.ibm.wmqfte.cred.keyfile** Java

   Nota: Antes de IBM MQ 9.3.1 e IBM MQ 9.3.0 Fix Pack 10, o nome dessa propriedade de sistema Java estava escrito incorretamente como **com.ibm.wqmfte.cred.keyfile**. Em IBM MQ 9.3.1 e IBM MQ 9.3.0 Fix Pack 10, o Managed File Transfer usa ambas as versões da propriedade de sistema Java para manter a compatibilidade com versões anteriores. Se ambas as propriedades do sistema Java forem configuradas, o valor da propriedade ortográfica corretamente **com.ibm.wmqfte.cred.keyfile** será usado.
2. Propriedades no agente, criador de logs, comandos e arquivos de propriedade de coordenação.
3. A propriedade **commonCredentialsKeyFile** no arquivo `installation.properties` ..

Para obter informações adicionais, consulte “Criptografando credenciais armazenadas no MFT” na página 583.

Por padrão, o comando **fteObfuscate** protege as credenciais com um sistema de proteção que não é compatível com MFT versões anteriores a IBM MQ 9.2 Para proteger os arquivos de configuração com o sistema de proteção de credenciais compatível com versões anteriores a IBM MQ 9.2, especifique o parâmetro **-sp 0** quando o comando **fteObfuscate** for executado.

IBM MQ Internet Pass-Thru

O arquivo de configuração IBM MQ Internet Pass-Thru (MQIPT) pode conter senhas que são usadas para acessar vários recursos.

Proteja senhas no arquivo de configuração MQIPT usando o comando **mqiptPW** .

O comando **mqiptPW** solicita que a senha a ser criptografada seja inserida, e retorna a senha criptografada Copie a senha criptografada no arquivo de configuração MQIPT .

Por padrão, o comando **mqiptPW** criptografa uma senha com a chave inicial padrão Para criptografar a senha com uma chave inicial específica, use o parâmetro **-sf** para especificar o caminho para o arquivo que contém a chave inicial.

Cuidado: A chave inicial padrão é a mesma para todas as instalações do IBM MQ Para proteger senhas com segurança, forneça uma chave inicial que seja exclusiva para sua instalação ao criptografar senhas.

Para obter mais informações, consulte Especificando a chave de criptografia de senha..

Se você especificar um arquivo de chave inicial ao criptografar a senha do repositório de chaves, também deverá especificar o mesmo arquivo de chave inicial quando o MQIPT for iniciado Os seguintes mecanismos podem ser usados para especificar o nome do arquivo-chave inicial, em ordem de prioridade:

1. O parâmetro **-sf** no comando usado para iniciar o MQIPT.
2. a variável de ambiente **MQS_MQIPTCRED_KEYFILE**.
3. a propriedade **com.ibm.mq.ipt.cred.keyfile** Java.
4. Um arquivo denominado `mqipt_cred.key` no diretório inicial do MQIPT O diretório inicial MQIPT é o diretório que contém o arquivo de configuração MQIPT .

Por padrão, o comando **mqiptPW** protege as credenciais com um sistema de proteção que não é compatível com MQIPT versões anteriores a IBM MQ 9.2 Para proteger as senhas com o sistema de proteção de credenciais que é compatível com versões anteriores a IBM MQ 9.2, use a sintaxe de comando **mqiptPW** que é suportada em versões anteriores a IBM MQ 9.2

IBM MQ clients que usam hardware criptográfico

É possível configurar clientes do IBM MQ para usar o hardware de criptografia PKCS #11 para armazenar chaves privadas e certificados que são usados em comunicações TLS. Para acessar dispositivos PKCS #11 , deve-se fornecer uma senha como parte da cadeia de configuração fornecida para o IBM MQ client.

Importante: As senhas fornecidas usando o campo **CryptoHardware** na estrutura MQSCO ou o atributo **SSLCRYP** do gerenciador de fila não podem ser protegidas usando esse mecanismo

É possível proteger essa senha usando o comando **runp11cred** , que pode ser localizado na pasta `bin` no diretório de instalação do IBM MQ

O comando **runp11cred** solicita que a senha a ser criptografada seja inserida, e retorna a senha criptografada A senha criptografada deve ser copiada na sequência de configuração de hardware de criptografia.

Por exemplo, se sua cadeia de configuração de hardware criptográfico for a seguinte::

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;Passw0rd;SYMMETRIC_CIPHER_ON
```

Quando o comando **runp11cred** solicitar que você insira a senha, insira `Passw0rd` O comando retorna uma sequência semelhante à seguinte:

```
<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==
```

Substitua a senha na sequência de configuração de hardware criptográfico pela sequência que é retornada pelo comando **runp11cred** , para fornecer a sequência a seguir que contém a senha criptografada:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHm SNF0/
ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

Quando o aplicativo IBM MQ client for executado, especifique a sequência de configuração de hardware de criptografia que contém a senha criptografada em um dos métodos a seguir:

- O atributo **SSLCryptoHardware** na sub-rotina SSL do arquivo de configuração do cliente..
- A variável de ambiente **MQSSLCRYP** .

Por padrão, o comando **runp11cred** criptografa uma senha com uma chave inicial padrão.. Para proteger uma senha com sua própria chave inicial, especifique o nome do arquivo que contém a chave inicial usando um dos mecanismos a seguir, em ordem de prioridade:

1. O parâmetro **-sf** para o comando **runp11cred**
2. a variável de ambiente **MQS_SSLCRYP_KEYFILE**.

Cuidado: A chave inicial padrão é a mesma para todas as instalações do IBM MQ Para proteger senhas com segurança, forneça uma chave inicial que seja exclusiva para sua instalação ao criptografar senhas.

Se você especificar um arquivo-chave inicial ao criptografar a senha do repositório de chaves, também deverá especificar o nome do arquivo que contém a chave inicial quando o IBM MQ client for executado Especifique o nome do arquivo-chave inicial usando um dos mecanismos a seguir, em ordem de prioridade:

1. A variável de ambiente **MQS_SSLCRYP_KEYFILE** .
2. O atributo **SSLCryptoHardwareKeyFile** na sub-rotina **SSL** do arquivo de configuração do cliente

Gerenciador de filas da IBM MQ

O gerenciador de filas IBM MQ armazena senhas internamente em vários atributos. Por exemplo, o atributo **KEYRPWD** do gerenciador de filas.. O gerenciador de filas criptografa automaticamente a senha antes que ela seja armazenada em arquivos no disco

A senha para o repositório de chaves TLS do gerenciador de filas pode ser protegida usando o sistema de proteção de senha do IBM MQ ou um arquivo stash do repositório de chaves Para obter mais informações sobre esses dois métodos, consulte “Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301

Quando o gerenciador de filas criptografa uma senha, a chave inicial padrão é usada a menos que você especifique sua própria chave inicial. Para usar sua própria chave inicial, configure o atributo **INITKEY** do gerenciador de fila para uma chave exclusiva e forte antes de configurar quaisquer atributos do gerenciador de filas criptografados

Cuidado: A chave inicial padrão é a mesma para todas as instalações do IBM MQ Para proteger senhas com segurança, forneça uma chave inicial que seja exclusiva para sua instalação ao criptografar senhas.

Aviso: Se a chave inicial for modificada depois de configurar o valor de atributos criptografados, os atributos criptografados não serão criptografados novamente com a nova chave inicial. Portanto, mudar a chave inicial sem reenviar a passphrase do repositório de chaves resulta em IBM MQ não conseguir decriptografar a passphrase do repositório de chaves e não conseguir acessar o repositório de chaves.

Para obter mais informações, consulte INITKEY..

IBM MQ C aplicativos clientes

As bibliotecas do cliente C do IBM MQ requerem senhas para acessar determinados recursos protegidos Por exemplo, um repositório de chaves TLS para aplicativos que usam TLS para conectar-se ao gerenciador de fila.

A senha do repositório de chave pode ser protegida usando o sistema de proteção de senha do IBM MQ ou um arquivo stash do repositório de chave. Para obter mais informações sobre esses dois métodos, consulte “Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301

Para proteger senhas com o sistema de proteção de senha IBM MQ , use o comando **runmqicred** . O comando está localizado no diretório *MQ_INSTALLATION_PATH*/bin

O comando **runmqicred** solicita que a senha a ser criptografada seja inserida, e retorna a senha criptografada A senha criptografada pode ser usada pelo aplicativo cliente em vez de uma senha de texto simples.

Por exemplo, se você optar por fornecer uma senha do repositório de chave TLS usando a variável de ambiente *MQKEYRPWD* e sua senha do keystore TLS for `Passw0rd`. Quando você executar **runmqicred**, insira `Passw0rd` quando solicitado O comando retorna uma sequência semelhante à seguinte:

```
<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w==
```

Configure essa sequência como o valor para a variável de ambiente *MQKEYRPWD* :

```
export MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
set MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
```

Por padrão, o comando **runmqicred** criptografa uma senha com a chave inicial padrão Para proteger uma senha com sua própria chave inicial, use um dos seguintes mecanismos para especificar o nome do arquivo que contém a chave, em ordem de prioridade:

1. O parâmetro **-sf** para o comando **runmqicred**
2. A variável de ambiente *MQS_MQI_KEYFILE* .

Cuidado: A chave inicial padrão é a mesma para todas as instalações do IBM MQ Para proteger senhas com segurança, forneça uma chave inicial que seja exclusiva para sua instalação ao criptografar senhas.

Se você especificar um arquivo-chave inicial ao criptografar a senha, também deverá tornar a chave inicial disponível para o aplicativo cliente quando ele for executado

Para obter informações adicionais, consulte “Fornecendo a senha do repositório de chaves para um IBM MQ MQI client on AIX, Linux, and Windows” na página 307.

Configurações de HA nativa

V 9.4.0

O tráfego de replicação de log de HA nativa entre instâncias pode ser criptografado usando TLS. Os certificados que são usados para proteger o tráfego de replicação de log são armazenados em um repositório de chaves especificado na sub-rotina **NativeHALocalInstance** do arquivo `qm.ini` .

A senha do repositório de chave pode ser protegida usando o sistema de proteção de senha do IBM MQ ou um arquivo stash do repositório de chave. Para obter mais informações sobre esses dois métodos, consulte “Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301

Para proteger a senha do repositório de chaves HA nativa com o sistema de proteção de senha IBM MQ , use o comando **runmqicred** .

O comando **runmqicred** solicita que a senha a ser criptografada seja inserida, e retorna a senha criptografada A senha criptografada deve ser usada em vez de uma senha de texto simples. Configure o valor do atributo **KeyRepositoryPassword** na sub-rotina **NativeHALocalInstance** do arquivo `qm.ini` para a senha criptografada que o comando retorna

Por padrão, o comando **runmqicred** criptografa uma senha com a chave inicial padrão Para proteger uma senha com sua própria chave inicial, use um dos seguintes mecanismos para especificar o nome do arquivo que contém a chave, em ordem de prioridade:

1. O parâmetro **-sf** para o comando **runmqicred**
2. A variável de ambiente *MQS_MQI_KEYFILE* .

Cuidado: A chave inicial padrão é a mesma para todas as instalações do IBM MQ Para proteger senhas com segurança, forneça uma chave inicial que seja exclusiva para sua instalação ao criptografar senhas.

Se você especificar um arquivo de chave inicial ao criptografar a senha de repositório de chaves, também deverá especificar o mesmo arquivo de chave inicial usando o atributo **InitialKeyFile** na sub-rotina **NativeHALocalInstance** do arquivo `qm.ini`

Para obter mais informações, consulte NativeHALocalSub-rotina da instância do arquivo qm.ini.

Gerenciador de filas IBM MQ (sub-rotinaAuthToken no arquivo `qm.ini`)

Linux V 9.4.0 AIX

Em IBM MQ 9.3.4, IBM MQ MQI clients que se conectam aos gerenciadores de filas do IBM MQ que são executados em sistemas AIX ou Linux , pode usar tokens de autenticação para autenticar com o gerenciador de filas. O gerenciador de filas deve ser configurado para aceitar tokens de autenticação e ser capaz de acessar o certificado de chave pública do emissor de token ou a chave secreta usada para assinar o token. O repositório de chaves que contém certificados de chave pública ou chaves secretas do emissor confiável é protegido com uma senha.

A senha do repositório de chave pode ser protegida usando o sistema de proteção de senha do IBM MQ ou um arquivo stash do repositório de chave. Para obter mais informações sobre esses dois métodos, consulte “Criptografando senhas do repositório de chaves no AIX, Linux, and Windows” na página 301

Para proteger a senha do repositório de chaves do token de autenticação com o sistema de proteção de senha do IBM MQ , use o comando **runqmcred** para criptografar a senha

O comando **runqmcred** solicita que a senha a ser criptografada seja inserida, e retorna a senha criptografada A senha criptografada deve ser usada em vez de uma senha de texto simples. Copie a senha criptografada em um arquivo e inclua o caminho no arquivo no atributo **KeyStorePwdFile** da sub-rotina **AuthToken** no arquivo `qm.ini` .

Por padrão, o comando **runqmcred** criptografa uma senha com a chave inicial padrão Para criptografar a senha com uma chave inicial específica, use o parâmetro **-sf** para especificar o caminho para o arquivo que contém a chave inicial.

Cuidado: A chave inicial padrão é a mesma para todas as instalações do IBM MQ Para proteger senhas com segurança, forneça uma chave inicial que seja exclusiva para sua instalação ao criptografar senhas.

Importante: Se você fornecer uma chave inicial quando criptografar a senha, a mesma chave inicial deverá ser especificada no atributo **INITKEY** do gerenciador de fila para que o gerenciador de filas possa decriptografar a senha Se o atributo **INITKEY** do gerenciador de filas já estiver configurado, use a mesma chave inicial quando você executar o comando **runqmcred** Para obter mais informações sobre o atributo **INITKEY** do gerenciador de fila, consulte INITKEY..

Por exemplo, para criptografar as senhas do keystore do token de autenticação com a chave inicial no arquivo `/home/initial.key`, emita o comando a seguir:

```
runqmcred -sf /home/initial.key
```

Para obter informações adicionais, consulte “Configurando um gerenciador de filas para aceitar tokens de autenticação usando um keystore local” na página 337.

Os limites para proteção por meio de criptografia de senha

O IBM MQ suporta a criptografia AES-128 para senhas armazenadas em vários arquivos de configuração. Ao usar a criptografia Advanced Encryption Standard (AES) para proteger as senhas na configuração do IBM MQ , é necessário entender os limites para a proteção que ele fornece

Criptografar uma senha nos arquivos de configuração IBM MQ não significa que a senha esteja segura ou protegida. Ele só impede que a senha seja facilmente recuperada por alguém que possa acessar a senha criptografada, mas não conhece a chave de criptografia. Os processos do IBM MQ requerem acesso à senha criptografada e à chave de decriptografia para obter a senha não criptografada para uso. Esses dois itens de dados devem ser armazenados no sistema de arquivos em um local acessível ao IBM MQ. Qualquer pessoa que criptografa uma senha colocada em um arquivo de configuração também requer acesso à chave de criptografia. Se um invasor tiver acesso ao mesmo conjunto de arquivos que o IBM MQ, aplicar a criptografia AES à senha, portanto, fornece apenas um nível mínimo de proteção.

No entanto, criptografar senhas em repouso é importante considerar, pois evita a divulgação acidental de senhas e permite o compartilhamento de arquivos de configuração, se a chave de descriptografia também não for compartilhada.

Além de assegurar que o arquivo que contém a chave de decriptografia não seja compartilhado, deve-se tomar cuidado para assegurar que o arquivo seja protegido de outros usuários no sistema Embora os arquivos de configuração do IBM MQ possam ser acessíveis a todos os usuários, restrinja as permissões no arquivo que contém a chave de decriptografia ao mínimo necessário Os IDs do usuário que os processos do IBM MQ executam devem ter acesso concedido para ler o arquivo que contém a chave de decriptografia. No entanto, não é necessário conceder acesso para ler o arquivo para um grupo ou todos os usuários no sistema.

Proteção de detalhes de autenticação do banco de dados

Se você estiver usando a autenticação de nome de usuário e senha para se conectar ao gerenciador do banco de dados, poderá armazená-los no armazenamento de credenciais do MQ XA para evitar armazenar a senha em texto sem formatação no arquivo `qm.ini`.

Atualize XAOpenString para o gerenciador de recursos

Para usar o armazenamento de credenciais, deve-se modificar o XAOpenString no arquivo `qm.ini`. A sequência é usada para se conectar ao gerenciador do banco de dados. É possível especificar campos substituíveis para identificar onde o nome de usuário e a senha são substituídos dentro da sequência XAOpenString.

- O campo +USER+ é substituído pelo valor do nome de usuário armazenado no armazenamento de XACredentials.
- O campo +PASSWORD+ é substituído pelo valor de senha armazenado no armazenamento de XACredentials..

Os exemplos a seguir mostram como modificar um XAOpenString para usar o arquivo de credenciais para se conectar ao banco de dados.

Conectando-se a um banco de dados Db2

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

Conectando-se a um banco de dados Oracle

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

Trabalhe com as credenciais para o banco de dados para o armazenamento das credenciais de MQ XA

Depois de atualizar o arquivo `qm.ini` com as sequências credenciais substituíveis, deve-se incluir o nome do usuário e a senha para o armazenamento de credenciais MQ usando o comando **setmqxacred**. Também é possível usar **setmqxacred** para modificar as credenciais existentes, excluir credenciais ou listar as credenciais. Os exemplos a seguir fornecem alguns casos de uso típico:

Incluindo Credenciais

O comando a seguir salva com segurança o nome do usuário e a senha para o gerenciador de filas `QM1` para o recurso `mqdb2`.

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

Atualizando credenciais

Para atualizar o nome do usuário e a senha usados para conectar a um banco de dados, emita novamente o comando **setmqxacred** com o novo nome do usuário e a senha:

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

Deve-se reiniciar o gerenciador de filas para que as mudanças entrem em vigor.

Excluindo credenciais

O seguinte comando exclui as credenciais:

```
setmqxacred -m QM1 -x mydb2 -d
```

Listando as credenciais

O seguinte comando lista as credenciais:

```
setmqxacred -m QM1 -l
```

Referências relacionadas

setmqxacred

Segurança do Managed File Transfer

Diretamente após a instalação e sem nenhuma modificação, o Managed File Transfer terá um nível de segurança que pode ser apropriado para propósitos de teste ou avaliação em um ambiente protegido. Entretanto, em um ambiente de produção, deve-se considerar o controle apropriado de quem pode iniciar as operações de transferência de arquivos, quem pode ler e gravar os arquivos que estão sendo transferidos e como proteger a integridade dos arquivos.

Tarefas relacionadas

Restringindo autoridades grupo para recursos específicos do MFT

Gerenciando autoridades para recursos específicos do MFT

“Usando o Advanced Message Security com o Managed File Transfer” na página 650

Este cenário explica como configurar o Advanced Message Security para fornecer privacidade de mensagem para dados que estão sendo enviados por meio de um Managed File Transfer.

Referências relacionadas

Autoridades para MFT para acessar sistemas de arquivos

Propriedade commandPath do MFT

Autoridade para publicar mensagens de log e de status dos agentes MFT

Criptografando credenciais armazenadas no MFT

Managed File Transfer (MFT) requer vários IDs de usuário e credenciais, que são armazenados em dois arquivos XML e é possível ofuscá-los usando o comando **fteObfuscate** .

Arquivos de credencial

MQMFTCredentials.xml

Esse arquivo contém o ID do usuário e as credenciais para conexão com agentes e gerenciadores de filas de comando e coordenação. As credenciais para acessar os keystores de conexões seguras com os gerenciadores de filas também são armazenadas no mesmo arquivo.

Consulte “Autenticação de conexão do MFT e IBM MQ” na página 587 para obter detalhes dos valores da propriedade que definem a localização do arquivo `MQMFTCredentials.xml` .

ProtocolBridgeCredentials.xml

Esse arquivo contém o ID do usuário e as credenciais para conexão com os servidores de protocolo.

Criptografando credenciais usando o comando `fteObfuscate`

O comando **`fteObfuscate`** aceita os seguintes parâmetros:

- **`-f`** *credentials_file_name* (necessário)

  Nota: Deprecated Esse parâmetro substitui o parâmetro **`-credentialsFile`** que foi descontinuado de IBM MQ 9.2.0.
- **`-sp`** *protection_mode*
- **`-sf`** *credentials_key_file*
- **`-o`** *output_file_name*

Consulte **`fteObfuscate`** para obter detalhes sobre os parâmetros.

Se você não especificar o modo de proteção ou um arquivo-chave de credenciais, o comando usará o modo de proteção padrão e o algoritmo mais recente, mas com uma chave fixa para criptografar as credenciais.

Se você especificar um modo de proteção de `0` e não especificar um arquivo-chave de credenciais, o comando funcionará da mesma forma que em liberações anteriores do produto. Você recebe uma mensagem de aviso no console indicando o uso de proteção descontinuada.

Se você especificar um modo de proteção de `0` e um arquivo-chave de credenciais, receberá uma saída de erro no console indicando que não é válido especificar o arquivo-chave ao usar o modo de proteção `0`.

Se você especificar o modo de proteção de 1 e não especificar um arquivo-chave de credenciais, o comando usará o algoritmo mais recente, mas com uma chave fixa para criptografar as credenciais.

Se você especificar o modo de proteção de 1 e especificar um arquivo-chave de credenciais, o comando criptografará as credenciais com o algoritmo mais recente.

Se você especificar o modo de proteção de 1 ou não especificar o modo de proteção, mas especificar um arquivo-chave de credenciais que não existe, um erro será gerado no console indicando que o arquivo não existe.

Se você especificar o modo de proteção de 1 ou não especificar o modo de proteção e especificar um arquivo-chave de credenciais que não é legível, um erro será gerado no console indicando que o arquivo não está legível.

Se você especificar o modo de proteção de 2e não especificar um arquivo-chave de credenciais, o comando usará o modo de proteção 2 para criptografar credenciais usando o algoritmo mais recente e uma chave fixa para criptografar...

Se você especificar o modo de proteção de 2e especificar um arquivo-chave de credenciais, o comando usará o modo de proteção 2 para criptografar credenciais usando o algoritmo mais recente e uma chave especificada pelo usuário para criptografar.

Se você especificar o modo de proteção de 2 ou não especificar o modo de proteção, mas especificar um arquivo-chave de credenciais que não existe, um erro será gerado no console indicando que o arquivo não existe.

Se você especificar o modo de proteção de 2 ou não especificar o modo de proteção e especificar um arquivo-chave de credenciais que não é legível, um erro será gerado no console indicando que o arquivo não está legível.

Decriptografando credenciais

É possível especificar o caminho para o arquivo-chave inicial em vários lugares. Para decriptografar credenciais que foram criptografadas usando uma chave inicial diferente da padrão, o nome do arquivo que contém a chave inicial precisa ser fornecido para MFT de uma das maneiras a seguir, nesta ordem de precedência:

1. Usando uma propriedade de sistema Java , por exemplo:

   ```
   -Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
   ```

   Nota:

   - Antes de IBM MQ 9.3.1 e IBM MQ 9.3.0 Fix Pack 10, o nome dessa propriedade de sistema Java estava escrito incorretamente no código do produto como `com.ibm.wqmfte.cred.keyfile`. Em IBM MQ 9.3.1 e IBM MQ 9.3.0 Fix Pack 10, a ortografia do nome da propriedade é corrigida para `com.ibm.wmqfte.cred.keyfile`. O Managed File Transfer usa ambas as versões da propriedade de sistema Java ao verificar se um usuário especificou um arquivo contendo a chave inicial que deve ser usada para criptografar e decriptografar credenciais. Isso permite que você use a ortografia correta do nome da propriedade, enquanto mantém a compatibilidade com versões anteriores com o nome digitado incorretamente. Observe que se ambas as propriedades do sistema Java forem configuradas, o valor da propriedade `com.ibm.wmqfte.cred.keyfile` corretamente escrita será usado.
   - Antes IBM MQ 9.3.1 e IBM MQ 9.3.0 Fix Pack 10, use a propriedade `com.ibm.wqmfte.cred.keyfile`.

   .
2. Configurando uma propriedade em um arquivo de propriedades de agente, comando, coordenação ou criador de log. O nome do arquivo de propriedades e a propriedade que precisa ser configurada nele são mostrados na tabela a seguir:

Arquivo de Propriedades	**Nome da Propriedade**
agent.properties	agentCredentialsKeyFile
command.properties	commandCredentialsKeyFile
coordination.properties	coordinationCredentialsKeyFile
logger.properties	loggerCredentialsKeyFile

3. No arquivo installation.properties.

   Em vez de incluir propriedades em arquivos de propriedades individuais, é possível incluir a propriedade **commonCredentialsKeyFile** no arquivo `installation.properties` comum existente, para que o agente, o criador de logs e os comandos possam usar a mesma propriedade.

Se você tiver definido as várias propriedades **CredentialsKeyFile** em vários locais:

- O caminho do arquivo-chave de credenciais que está sendo usado para o agente e o criador de logs é registrado no arquivo `output0.log` para esse agente ou criador de logs
- O caminho do arquivo-chave de credenciais sendo usado para os comandos é exibido no console.

A Java propriedade de sistema **com.ibm.wmqfte.cred.keyfile** substitui todos os outros.. Se a propriedade do sistema não estiver configurada, o agente procurará o arquivo `agent.properties`, seguido pelo arquivo `installation.properties` do arquivo-chave inicial.

Se o arquivo de chave inicial ainda não for localizado, e você tiver configurado o modo de proteção no comando **fteObfuscate** para 1, o agente registrará uma mensagem de erro no arquivo `output0.log` .

Se você tiver configurado o modo de proteção como `0` no comando **fteObfuscate**, uma mensagem de aviso será registrada indicando a descontinuação.

O criador de logs e os comandos seguem as mesmas etapas para localizar o arquivo-chave inicial.

Ponte de Protocolo e Ponte Connect:Direct

O Bridge Protocol usa um arquivo de propriedades, `ProtocolBridgeProperties.xml`, para conexão com servidores FTP, SFTP e FTPS. Esse arquivo de propriedades contém atributos de conexão necessários para se conectar a esses servidores.

Um reinício do agente de ponte será necessário se você modificar o valor dos atributos **credentialsFile** ou **credentialsKeyFile** no arquivo `ProtocolBridgeProperties.xml`.

Um dos atributos é o **credentialsFile** e o valor dele contém o caminho para um arquivo XML contendo o UID, o PWD ou a Chave necessária para se conectar a esses servidores. O valor padrão para o atributo é *ProtocolBridgeCredentials.xml* e o arquivo está em seu diretório inicial, exatamente como o arquivo `MQMFTCredentials.xml` .

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

Assim como *MQMFTCredentails.xml*, é possível criptografar *ProtocolBridgeCredentials.xml* com o comando **fteObfuscate**. Para fins de decriptografia, é possível especificar o caminho necessário para um arquivo-chave de credenciais usando o elemento adicional **credentialsKeyFile** conforme mostrado no texto a seguir. O caminho pode conter variáveis de ambiente.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Nota: Especificar um valor para a propriedade do agente **agentCredentialsKeyFile** , a propriedade **commonCredentialsKeyFile** no `installation.propertiesou` por meio da propriedade de sistema **com.ibm.wqmfte.cred.keyfile**não tem nenhum impacto no valor especificado para o atributo **credentialsKeyFile** .

Da mesma forma, a Ponte Connect:Direct usa o *ConnectDirectNodeProperties.xml* para se conectar ao servidor Connect:Direct. O arquivo XML contém as informações de conexão necessárias, além de um atributo que define o caminho para o arquivo XML de credenciais. Esse arquivo XML de credenciais contém o ID do usuário ou o PWD, bem como informações adicionais necessárias para se conectar ao servidor Connect:Direct.

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

Assim como o arquivo *ProtocolBridgeCredentials.xml*, é possível criptografar *ConnectDirectCredentials.xml* com o comando **fteObfuscate**. Para fins de decriptografia, é possível especificar o caminho necessário para um arquivo-chave de credenciais usando o elemento adicional **credentialsKeyFile** conforme mostrado no texto a seguir. O caminho pode conter variáveis de ambiente.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Nota: Especificar um valor para a propriedade do agente **agentCredentialsKeyFile** e para a propriedade **commonCredentialsKeyFile** no `installation.properties` ou por meio da propriedade do sistema **com.ibm.wqmfte.cred.keyfile** não tem nenhum impacto no valor especificado para o atributo **credentialsKeyFile**.

É possível especificar o elemento **credentialsKeyFile**, sem especificar o elemento **credentialsFile** no arquivo *ProtocolBridgeProperties.xml*.

Se você não especificar o elemento **credentialsFile**, o arquivo de credencial padrão *ProtocolBridgeCredentials.xml* será usado pelo agente de ponte de protocolo e o valor do arquivo-chave especificado no atributo **credentialsKeyFile** será usado para decriptografar o arquivo de credencial.

Da mesma forma, é possível especificar o elemento **credentialsKeyFile**, sem especificar o elemento **credentialsFile** no arquivo *ConnectDirectNodeProperties.xml*.

Se você não especificar o elemento **credentialsFile**, o arquivo de credencial padrão *ConnectDirectCredentials.xml* será usado pela ponte Connect:Direct e o valor do arquivo-chave especificado no atributo **credentialsKeyFile** será usado para decriptografar o arquivo de credencial.

Usando a chave do conjunto de dados no z/OS

No z/OS, é possível especificar o **MQMFTCredentials** e fornecer o arquivo-chave de credenciais usando um PDSE. Consulte “Configuring MQMFTCredentials.xml on z/OS” na página 589.

Referências relacionadas

Qual Comando do MFT se Conecta a qual Gerenciador de Filas

Formato de arquivo de credenciais do MFT

fteObfuscate (criptografar dados sensíveis)

Autenticação de conexão do MFT e IBM MQ

A autenticação de conexão permite que um gerenciador de filas seja configurado para autenticar aplicativos usando um ID do usuário e uma senha fornecidos. Se o gerenciador de filas associado tiver a segurança ativada e requerer detalhes da credencial (ID do usuário e senha), o recurso de autenticação de conexão deverá ser ativado antes que uma conexão bem-sucedida com um gerenciador de filas possa ser feita. A autenticação de conexão pode ser executada no modo de compatibilidade ou no modo de autenticação MQCSP.

Métodos de Fornecer Detalhes da Credencial

Muitos comandos Managed File Transfer suportam os seguintes métodos de fornecimento de detalhes de credencial:

Detalhes fornecidos por argumentos da linha de comandos.
Os detalhes da credencial podem ser especificados usando os parâmetros **-mquserid** e **-mqpassword** . Se o **-mqpassword** não for fornecido, então, o usuário será solicitado a fornecer a senha em que a entrada não é exibida.

Detalhes fornecidos por meio de um arquivo de credenciais: `MQMFTCredentials.xml`.
Os detalhes da credencial podem ser predefinidos em um arquivo `MQMFTCredentials.xml` como texto não criptografado ou texto ofuscado.

Multi Para obter informações sobre como configurar um arquivo MQMFTCredentials.xml em IBM MQ for Multiplatforms consulte “Configurando MQMFTCredentials.xml em Multiplataformas” na página 588.

z/OS Para obter informações sobre como configurar um arquivo MQMFTCredentials.xml em IBM MQ for z/OS consulte “Configuring MQMFTCredentials.xml on z/OS” na página 589.

Precedência

A precedência de determinar os detalhes de credencial é:

1. Argumento da linha de comandos.
2. Índice do `MQMFTCredentials.xml` por Gerenciador de Filas associado e usuário executando o comando.
3. Índice do `MQMFTCredentials.xml` por Gerenciador de Filas associado.
4. Modo de compatibilidade com versões anteriores padrão em que nenhum detalhe de credencial é fornecido para permitir compatibilidade com liberações anteriores do IBM MQou IBM WebSphere MQ

Notas:

- Os comandos **`fteStartAgent`** e **`fteStartLogger`** não suportam o argumento de linha de comandos **-mquserid**, ou **-mqpassword**e os detalhes de credencial só podem ser especificados com o arquivo `MQMFTCredentials.xml`.
- z/OS

  No z/OS, a senha deve ser maiúscula, mesmo se a senha do usuário tiver letras minúsculas. Por exemplo, se a senha do usuário foi "senha", terá que ser inserida como "SENHA".

Referências relacionadas

Qual Comando do MFT se Conecta a qual Gerenciador de Filas

Formato de arquivo de credenciais do MFT

Configurando MQMFTCredentials.xml em Multiplataformas

Se o Managed File Transfer (MFT) for configurado com a segurança ativada, a autenticação de conexão requererá todos os comandos do MFT que se conectam a um gerenciador de filas para fornecer credenciais de ID do usuário e senha Da mesma forma, os criadores de logs do MFT podem ser necessários para especificar um ID do usuário e senha ao se conectar a um banco de dados. Essas informações de credenciais podem ser armazenadas no arquivo de credenciais MFT .

Sobre esta tarefa

Os elementos no arquivo `MQMFTCredentials.xml` devem estar em conformidade com o esquema `MQMFTCredentials.xsd`. Para obter informações sobre o formato de `MQMFTCredentials.xml`, consulte Formato de arquivo de credenciais do MFT.

É possível localizar um arquivo de credenciais de amostra no diretório `MQ_INSTALLATION_PATH/mqft/samples/credentials` .

Você pode ter um MFT arquivo de credenciais para o gerenciador de filas de coordenação, um para o gerenciador de filas de comando, um para cada agente e um para cada logger. Como alternativa, é possível ter um arquivo que seja usado por tudo em sua topologia

O local padrão do arquivo de credenciais MFT é o seguinte:

AIX and Linux

$HOME

Windows

%USERPROFILE% ou %HOMEDRIVE%%HOMEPATH%

Se o arquivo de credenciais estiver armazenado em um local diferente, será possível usar as propriedades a seguir para especificar onde os comandos devem procurá-lo:

Tabela 97. : Propriedades que definem o local do arquivo MQMFTCredentials.xml para vários comandos.

Tipo de comando	Arquivo de Propriedades	Nome da Propriedade
Comando que se conecta ao gerenciador de filas de coordenação	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
Comando que se conecta ao gerenciador de fila de comandos	connection.properties	connectionQMgrAuthenticationCredentialsFile
Comando que se conecta a um processo do agente	agent.properties	agentQMgrAuthenticationCredentialsFile
Comando que se conecta a um processo do criador de logs	logger.properties	loggerQMgrAuthenticationCredentialsFile

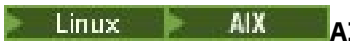
Tabela 98. : Propriedades que definem o local do arquivo MQMFTCredentials.xml para agentes e processos do criador de logs..

Tipo de comando	Arquivo de Propriedades	Nome da Propriedade
Agentes do MFT	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT criadores de logs	logger.properties	loggerQMgrAuthenticationCredentialsFile

Para obter detalhes sobre quais comandos e processos se conectam a qual gerenciador de filas, consulte Quais MFT Comandos e Processos se Conectam a Qual Gerenciador de Filas

Em vez de incluir propriedades em arquivos de propriedades individuais, é possível incluir a propriedade **commonCredentialsKeyFile** no arquivo `installation.properties` comum existente, de modo que o agente, o criador de logs e os comandos possam usar a mesma propriedade

Como o arquivo de credenciais contém informações de ID do usuário e senha, ele requer permissões especiais para evitar o acesso não autorizado a ele:



AIX and Linux

```
chown <agent owner userid>
chmod 600
```

Windows

Assegure-se de que a herança não esteja ativada e, em seguida, remova todos os IDs do usuário, exceto aqueles que estão executando o agente ou criador de logs que estarão usando o arquivo de credenciais.

Os detalhes da credencial usados para se conectar a um gerenciador de fila de coordenação do MFT , no plug-in do IBM MQ Explorer Managed File Transfer , dependem do tipo de configuração:

Global (configuração no disco local)

Uma configuração global usa o arquivo de credenciais especificado nas propriedades de coordenação e de comando.

Local (definido em IBM MQ Explorer):

Uma configuração local usa as propriedades dos detalhes de conexão do gerenciador de filas associado no IBM MQ Explorer.

Tarefas relacionadas

“Ativando a autenticação de conexão para o MFT” na página 591
A autenticação de conexão do Plug-in do IBM MQ Explorer MFT conectando-se a um gerenciador de filas de coordenação ou a um gerenciador de filas de comando e a autenticação de conexão para um agente do Managed File Transfer conectando-se a um gerenciador de filas de coordenação ou a um gerenciador de filas de comando podem ser executadas no modo de compatibilidade ou no modo de autenticação do MQCSP.

Criando uma estrutura de transferência de arquivos do IBM MQ

Referências relacionadas

Formato de arquivo de credenciais do MFT

Criptografando credenciais armazenadas no MFT

fteObfuscate: criptografar dados sensíveis

Configuring `MQMFTCredentials.xml` on z/OS

If Managed File Transfer (MFT) is configured with security enabled, connection authentication requires all MFT agents, and commands that connect to a queue manager, to supply user ID and password credentials.

Similarly, MFT loggers might be required to specify a user ID and password when connecting to a database.

This credential information can be stored in the MFT credentials file. Note that the credentials files are optional, however, it is easier to define the file or files that you require before you customize the environment.

In addition to this, if you have credentials files, you receive fewer warning messages. The warning messages inform you that MFT considers that queue manager security is off, and therefore you are not supplying authentication details.

You can find a sample credentials file in the `MQ_INSTALLATION_PATH/mqft/samples/credentials` directory.

Here is an example of an `MQMFTCredentials.xml`file:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MFTCredentials.xsd">
  <tns:qmgr name="MQPH" user="ADMIN" mqUserId="JOHNDOEH" mqPassword="cXXXX" />
  <tns:qmgr name="MQPI" user="ADMIN" mqUserId="JOHNDOEI" mqPassword="yXXXX" />
  <tns:qmgr name="MQPH" mqUserId="NONEH" mqPassword="yXXXX" />
  <tns:qmgr name="MQPI" mqUserId="NONEI" mqPassword="yXXXX" />
</tns:mqmftCredentials>
```

When a job with userid ADMIN needs to connect to queue manager MQPH, it passes user ID *JOHNDOEH* and uses password *cXXXX*.

If the job is run by any other user ID, and connects MQPH, that job passes user ID *NONEH* and password *yXXXX*.

The default location for the `MQMFTCredentials.xml` file is the user's home directory on z/OS UNIX System Services (USS). It is also possible to store the file in either a different location on USS, or in a member within a partitioned data set.

If the credentials file is stored in a different location, then you can use the following properties to specify where the commands should look for it:

Table 99. : Properties that define the location of the MQMFTCredentials.xml file for various commands.

Type of command	**Property file**	**Property name**
Command which connects to the coordination queue manager	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
Command which connects to the command queue manager	connection.properties	connectionQMgrAuthenticationCredentialsFile
Command that connects to an agent process	agent.properties	agentQMgrAuthenticationCredentialsFile
Command that connects to a logger process	logger.properties	loggerQMgrAuthenticationCredentialsFile

Table 100. : Properties that define the location of the MQMFTCredentials.xml file for agents and logger processes.

Type of command	**Property file**	**Property name**
MFT agents	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT loggers	logger.properties	loggerQMgrAuthenticationCredentialsFile

For details about what commands and processes connect to which queue manager, see Which MFT commands and processes connect to which queue manager.

To create the credentials file within a partitioned data set, carry out the following steps:

- Create a PDSE with format VB and logical record length (Lrecl) 200.
- Create a member within the data set, make a note of the data set and member, and add the following code to the member:

```
<?xml version="1.0" encoding="IBM-1047"?>
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/MFTCredentials MQMFTCredentials.xsd">
  <!—-credentials information goes here-->
</tns:mqmftCredentials>
```

You can protect the credentials file using a security product, for example, RACF, but the user IDs running the Managed File Transfer commands, and administering the agent and logger processes, need read access to this file.

You can obscure information in this file using the JCL in member BFGCROBS. This takes the file and encrypts the IBM MQ user ID and password. For example member BFGCROBS takes the line

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

and creates

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

If you want to keep the user ID to IBM MQ user ID mapping, you can add comments to the file. For example

```
<!--      name="MQPI" user="ADMIN"    mqUserId="JOHNDOE1 -->
```

These comments are unchanged by the obscuring process.

Note that the content is obscured, not strongly encrypted. You should limit which user IDs have access to the file.

Related tasks

"Configurando MQMFTCredentials.xml em Multiplataformas" on page 588
Se o Managed File Transfer (MFT) for configurado com a segurança ativada, a autenticação de conexão requererá todos os comandos do MFT que se conectam a um gerenciador de filas para fornecer credenciais de ID do usuário e senha Da mesma forma, os criadores de logs do MFT podem ser necessários para especificar um ID do usuário e senha ao se conectar a um banco de dados. Essas informações de credenciais podem ser armazenadas no arquivo de credenciais MFT .

Ativando a autenticação de conexão para o MFT

A autenticação de conexão do Plug-in do IBM MQ Explorer MFT conectando-se a um gerenciador de filas de coordenação ou a um gerenciador de filas de comando e a autenticação de conexão para um agente do Managed File Transfer conectando-se a um gerenciador de filas de coordenação ou a um gerenciador de filas de comando podem ser executadas no modo de compatibilidade ou no modo de autenticação do MQCSP.

Sobre esta tarefa

O modo de autenticação do MQCSP é o padrão

Para a autenticação de conexão para o plug-in do IBM MQ Explorer Managed File Transfer ou para os agentes do Managed File Transfer que se conectam a um gerenciador de filas usando o transporte CLIENT, as senhas com mais de 12 caracteres são suportadas somente para o modo de autenticação do MQCSP. Se você especificar uma senha com mais de 12 caracteres de comprimento ao autorizar o uso do modo de compatibilidade, ocorrerá um erro e o agente não será autenticado com o gerenciador de filas. Veja a mensagem BFGAG0187E em Mensagens de diagnóstico: BFGAG0001 - BFGAG9999.

Procedimento

- Para selecionar o modo de autenticação de conexão para um gerenciador de filas de coordenação ou para um gerenciador de filas de comandos no IBM MQ Explorer, conclua as etapas a seguir:

a) Selecione o gerenciador de filas ao qual você deseja se conectar.

b) Clique com o botão direito do mouse e selecione **Detalhes da Conexão-> Propriedades** no menu pop-up.

c) Clique na guia **ID do usuário**.

d) Certifique-se de que a caixa de seleção para o modo de autenticação de conexão que você deseja usar esteja selecionada:

   – Por padrão, a caixa de seleção **Modo de compatibilidade de identificação do usuário** é desmarcada. Isso significa que se a caixa de seleção **Ativar identificação de usuário** for selecionada, o IBM MQ Explorer usará a autenticação do MQCSP ao conectar-se ao gerenciador de filas. Se o IBM MQ Explorer precisar se conectar ao gerenciador de filas usando o modo de compatibilidade ao invés da autenticação do MQCSP, você deverá assegure-se de que as duas caixas de seleção **Ativar identificação de usuário** e **Modo de compatibilidade de identificação de usuário** estejam selecionadas.

- Para ativar ou desativar o modo de autenticação do MQCSP para um agente do Managed File Transfer usando o arquivo `MQMFTCredentials.xml`, inclua o parâmetro **useMQCSPAuthentication** no arquivo `MQMFTCredentials.xml` para o usuário relevante.

  O parâmetro **useMQCSPAuthentication** tem os valores a seguir:

  true
  O modo de autenticação do MQCSP é usado para autenticar o usuário com o gerenciador de filas.

  `true` é o valor padrão. Se o parâmetro **useMQCSPAuthentication** não for especificado, ele será, por padrão, configurado como `true` e o modo de autenticação do MQCSP será usado para autenticar o usuário com o gerenciador de filas.

  false
  O modo de compatibilidade é usado para autenticar o usuário com o gerenciador de filas.

  O exemplo a seguir mostra como configurar o parâmetro **useMQCSPAuthentication** no arquivo `MQMFTCredentials.xml`:

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

Conceitos relacionados

“Proteção de senha do MQCSP” na página 32
As credenciais de autenticação que são especificadas na estrutura MQCSP podem ser protegidas usando o recurso de proteção de senha do MQCSP IBM MQ ou criptografadas usando a criptografia TLS.

Referências relacionadas

“Autenticação de conexão do MFT e IBM MQ” na página 587
A autenticação de conexão permite que um gerenciador de filas seja configurado para autenticar aplicativos usando um ID do usuário e uma senha fornecidos. Se o gerenciador de filas associado tiver a segurança ativada e requerer detalhes da credencial (ID do usuário e senha), o recurso de autenticação de conexão deverá ser ativado antes que uma conexão bem-sucedida com um gerenciador de filas possa ser feita. A autenticação de conexão pode ser executada no modo de compatibilidade ou no modo de autenticação MQCSP.

Formato de arquivo de credenciais do MFT

Ambientes de simulação do MFT

É possível restringir a área do sistema de arquivos que o agente pode acessar como parte de uma transferência. A área à qual o agente está restrito é chamada de ambiente de simulação. É possível aplicar restrições ao agente ou ao usuário que solicitar uma transferência.

Os ambientes de simulação não são suportados quando o agente é um agente de ponte de protocolo ou um agente de ponte Connect:Direct. Não é possível usar a criação de ambiente de simulação de agente para agentes que precisam transferir para ou a partir de filas do IBM MQ.

Referências relacionadas

“Trabalhando com ambientes de simulação do agente MFT” na página 593
Para incluir um nível de segurança adicional no Managed File Transfer, é possível restringir a área de um sistema de arquivos que um agente pode acessar.

“Trabalhando com ambientes de simulação do usuário do MFT” na página 594
É possível restringir a área do sistema de arquivos à qual os arquivos podem ser transferidos para/de com base no nome de usuário MQMD do usuário que solicita a transferência.

Trabalhando com ambientes de simulação do agente MFT

Para incluir um nível de segurança adicional no Managed File Transfer, é possível restringir a área de um sistema de arquivos que um agente pode acessar.

Não é possível usar a criação de ambiente de simulação de agente para os agentes que transferem para ou a partir de filas do IBM MQ. A restrição de acesso às filas do IBM MQ com a criação de ambiente de simulação pode ser implementada em vez de usar a criação de ambiente de simulação do usuário, que é a solução recomendada para quaisquer requisitos de criação de ambiente de simulação. Para obter mais informações sobre a criação de ambiente de simulação do usuário, consulte “Trabalhando com ambientes de simulação do usuário do MFT” na página 594

Para ativar a criação de ambiente de simulação do agente, inclua a propriedade a seguir no arquivo `agent.properties` para o agente que deseja restringir:

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

em que:

- *`restricted_directory_name`* é um caminho de diretório a ser permitido ou negado.
- `!` é opcional e especifica que o valor a seguir para *`restricted_directory_name`* é negado (excluído). Se `!` não for especificado, *`restricted_directory_name`* será um caminho permitido (incluído).
- *`separator`* é o separador específico da plataforma.

Por exemplo, se você deseja restringir o acesso que AGENT1 tem apenas ao diretório `/tmp`, mas não permitir que o subdiretório `private` seja acessado, configure a propriedade da seguinte forma no arquivo `agent.properties` pertencente a AGENT1: `sandboxRoot=/tmp:!/tmp/private`.

A propriedade sandboxRoot é descrita em Propriedades Avançadas do Agente

Ambas as criações de ambiente de simulação, do agente e do usuário, não são suportadas em agentes de ponte de protocolo ou em agentes de ponte do Connect:Direct.

Trabalhando em um ambiente de simulação em plataformas AIX, Linux, and Windows

Nas plataformas AIX, Linux, and Windows, a criação de ambiente de simulação restringe quais diretórios um Managed File Transfer Agent pode ler e escrever. Quando a criação de ambiente de simulação está ativada, o Managed File Transfer Agent pode ler e gravar nos diretórios especificados, conforme o permitido, e nos subdiretórios contidos nos diretórios especificados, a menos que os subdiretórios estejam especificados como negados no sandboxRoot. A criação de ambiente de simulação do Managed File Transfer não tem precedência sobre a segurança do sistema operacional. O usuário que iniciou o Managed File Transfer Agent deve ter o acesso apropriado no nível do sistema operacional a qualquer diretório para poder ler do diretório ou gravar nele. Um link simbólico para um diretório não será seguido se o diretório vinculado estiver fora dos diretórios sandboxRoot especificados (e subdiretórios).

Trabalhando em um Ambiente de Simulação no z/OS

z/OS No z/OS, a criação de ambiente de simulação restringe de quais qualificadores de nome do conjunto de dados o Managed File Transfer Agent pode ler e nos quais pode gravar. O usuário que iniciou o Managed File Transfer Agent deve ter as autoridades corretas do sistema operacional para quaisquer conjuntos de dados envolvidos. Se você colocar um valor do qualificador de nome do conjunto de dados sandboxRoot entre aspas duplas, o valor seguirá a convenção normal do z/OS e será tratado como completo. Se você omitir as aspas duplas, sandboxRoot será prefixado com o ID do usuário atual. Por exemplo, se você configurar a propriedade sandboxRoot para o seguinte: `sandboxRoot=//test`, o agente poderá acessar os seguintes conjuntos de dados (em notação z/OS padrão) `//`*`username`*`.test.**` No tempo de execução, se os níveis iniciais do nome do conjunto de dados totalmente resolvido não corresponderem a sandboxRoot, a solicitação de transferência será rejeitada.

Trabalhando em um Ambiente de Simulação em Sistemas IBM i

IBM i Para arquivos no sistema de arquivos integrado em sistemas IBM i, a criação de ambiente de simulação restringe de quais diretórios um Managed File Transfer Agent pode ler e nos quais pode gravar. Quando a criação de ambiente de simulação está ativada, o Managed File Transfer Agent pode ler e gravar nos diretórios especificados, conforme o permitido, e nos subdiretórios contidos nos diretórios especificados, a menos que os subdiretórios estejam especificados como negados no sandboxRoot. A criação de ambiente de simulação do Managed File Transfer não tem precedência sobre a segurança do sistema operacional. O usuário que iniciou o Managed File Transfer Agent deve ter o acesso apropriado no nível do sistema operacional a qualquer diretório para poder ler do diretório ou gravar nele. Um link simbólico para um diretório não será seguido se o diretório vinculado estiver fora dos diretórios sandboxRoot especificados (e subdiretórios).

Referências relacionadas

“Verificações adicionais para transferências curingas” na página 597
Se um agente tiver sido configurado com um ambiente de simulação do usuário ou do agente para restringir os locais para os quais e dos quais o agente pode transferir arquivos, é possível especificar se verificações adicionais devem ser feitas em transferências curingas para esse agente.

“Trabalhando com ambientes de simulação do agente MFT” na página 593
Para incluir um nível de segurança adicional no Managed File Transfer, é possível restringir a área de um sistema de arquivos que um agente pode acessar.

O arquivo MFT `agent.properties`

Trabalhando com ambientes de simulação do usuário do MFT

É possível restringir a área do sistema de arquivos à qual os arquivos podem ser transferidos para/de com base no nome de usuário MQMD do usuário que solicita a transferência.

Os ambientes de simulação não serão suportados quando o agente for um agente de ponte de protocolo ou um agente de ponte Connect:Direct.

Para ativar a criação de ambiente de simulação do agente, inclua a propriedade a seguir no arquivo `agent.properties` para o agente que deseja restringir:

```
userSandboxes=true
```

Quando esta propriedade está presente e configurada como true, o agente usa as informações no arquivo *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name*`/agents/`*agent_name*`/UserSandboxes.xml` para determinar quais partes do sistema de arquivos podem ser acessadas pelo usuário que solicita a transferência.

O XML `UserSandboxes.xml` é composto de um elemento `<agent>` que contém zero ou mais elementos `<sandbox>`. Esses elementos descrevem quais regras são aplicadas a quais usuários. O atributo `user` do elemento `<sandbox>` é um padrão usado para correspondência com o usuário MQMD da solicitação.

O arquivo `UserSandboxes.xml` é recarregado periodicamente pelo agente e quaisquer mudanças válidas no arquivo afetarão o comportamento do agente. O intervalo de recarregamento padrão é de 30 segundos. Este intervalo pode ser alterado especificando a propriedade do agente xmlConfigReloadInterval no arquivo `agent.properties`.

Se você especificar o atributo ou valor `userPattern="regex"`, o atributo `user` será interpretado como uma expressão regular Java. Para obter mais informações, consulte Expressões regulares usadas pelo MFT.

Se você não especificar o atributo `userPattern="regex"` ou o valor, o atributo `user` será interpretado como um padrão com os seguintes caracteres curinga:

- asterisco (*), que representa zero ou mais caracteres
- ponto de interrogação (?), que representa exatamente um caractere

As correspondências são realizadas na ordem em que os elementos `<sandbox>` estão listados no arquivo. Apenas a primeira correspondência é usada, todas as possíveis correspondências seguintes no arquivo são ignoradas. Se nenhum dos elementos `<sandbox>` especificados no arquivo corresponder ao usuário MQMD associado à mensagem de solicitação de transferência, a transferência não poderá acessar o sistema de arquivos. Quando uma correspondência foi encontrada entre o nome de usuário MQMD e um atributo `user`, a correspondência identifica um conjunto de regras dentro de um elemento `<sandbox>` que são aplicadas na transferência. Este conjunto de regras é usado para determinar quais arquivos ou conjuntos de dados, pode ser lido ou gravado como parte da transferência.

Cada conjunto de regras pode especificar um elemento `<read>`, que identifica quais arquivos podem ser lidos, e um elemento `<write>` que identifica quais arquivos podem ser gravados. Se você omitir os elementos `<read>` ou `<write>` de um conjunto de regras, presume-se que o usuário associado a esse conjunto de regras não tenha permissão para realizar leituras ou gravações, conforme apropriado.

Nota: O elemento `<read>` deve estar antes do elemento `<write>` e o elemento `<include>` deve estar antes do elemento `<exclude>` no arquivo `UserSandboxes.xml`.

Cada elemento `<read>` ou `<write>` contém um ou mais padrões que são usados para determinar se um arquivo está no ambiente de simulação e pode ser transferido. Especifique esses padrões usando os elementos `<include>` e `<exclude>`. O atributo `name` do elemento `<include>` ou `<exclude>` especifica o padrão a ser correspondido. Um atributo `type` opcional especifica se o valor do nome é um padrão de arquivo ou de fila. Se o atributo `type` não for especificado, o agente tratará o padrão como um padrão de caminho de arquivo ou diretório. Por exemplo:

```
<tns:read>
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

Os padrões `<include>` e `<exclude>` `name` são usados pelo agente para determinar se os arquivos, conjuntos de dados ou filas podem ser lidos ou gravados. Uma operação é permitida se o caminho de arquivo canônico, conjunto de dados ou nome da fila corresponder a pelo menos um dos padrões incluídos e exatamente zero dos padrões excluídos. Os padrões especificados usando o atributo `name` dos elementos `<include>` e `<exclude>` usam os separadores de caminho e as convenções apropriadas para a plataforma na qual o agente está em execução Se você especificou caminhos de arquivo relativos, os caminhos serão resolvidos em relação à propriedade `transferRoot` do agente.

Quando você especificar uma restrição de fila, uma sintaxe de QUEUE@QUEUEMANAGER será suportada com as seguintes regras:

- Se o caractere (@) estiver ausente da entrada, o padrão será tratado como um nome da fila que pode ser acessado em qualquer gerenciador de filas. Por exemplo, se o padrão for `name`, ele será tratado da mesma forma que `name@**`.
- Se o caractere (@) for o primeiro caractere na entrada, o padrão será tratado como um nome do gerenciador de filas e todas as filas no gerenciador de filas poderão ser acessadas. Por exemplo, se o padrão for `@name`, ele será tratado da mesma forma que `**@name`.

Os seguintes caracteres curinga têm significado especial quando você os especifica como parte do atributo name dos elementos <include> e <exclude> :

*****
Um único asterisco corresponde a zero ou mais caracteres em um nome de diretório ou em um qualificador de um nome do conjunto de dados ou nome da fila .

?
Um ponto de interrogação corresponde exatamente a um caractere em um nome de diretório ou em um qualificador de um nome do conjunto de dados ou nome da fila .

******
Dois caracteres de asterisco correspondem a zero ou mais nomes de diretórios ou zero ou mais qualificadores em um nome do conjunto de dados ou nome da fila. Além disso, os caminhos que terminam com um separador de caminhos possuem um "**" implícito incluído no final do caminho. Assim, /home/user/ é o mesmo que /home/user/**.

Por exemplo:

- O /**/test/** corresponde a qualquer arquivo que tenha um diretório test em seu caminho
- O /test/file? corresponde a qualquer arquivo dentro do diretório /test que começa com a sequência file seguida por qualquer caractere único
- O c:\test*.txt corresponde a qualquer arquivo dentro do diretório c:\test com uma extensão .txt
- O c:\test***.txt corresponde a qualquer arquivo dentro do diretório 'c:\test ou um de seus subdiretórios que tem uma extensão .txt
- z/OS O //'TEST.*.DATA' corresponde a qualquer conjunto de dados que tenha o primeiro qualificador de TEST, qualquer segundo qualificador e um terceiro qualificador de DATA.
- *@QM1 corresponde a qualquer fila no gerenciador de filas QM1 que possui um único qualificador...
- O TEST.*.QUEUE@QM1 corresponde a qualquer fila no Gerenciador de Filas QM1 que tem o primeiro qualificador de TEST, qualquer segundo qualificador e um terceiro qualificador de QUEUE.
- **@QM1 corresponde a qualquer fila no gerenciador de filas QM1.

Links Simbólicos

Deve-se resolver completamente qualquer link simbólico que você usa nos caminhos de arquivo no arquivo UserSandboxes.xml especificando os links de disco rígido nos elementos <include> e <exclude>. Por exemplo, se você tiver um link simbólico no qual /var é mapeado para /SYSTEM/var, deverá especificar esse caminho como <tns:include name="/SYSTEM/var"/>, caso contrário, a transferência desejada falhará com um erro de segurança do ambiente de simulação do usuário.

exemplo

Este exemplo mostra como permitir que o usuário com o nome de usuário do MQMD guest transfira qualquer arquivo do diretório /home/user/public ou qualquer um de seus subdiretórios no sistema no qual o agente AGENT_JUPITER está em execução, incluindo o elemento <sandbox> a seguir no arquivo UserSandboxes.xml no diretório de configuração do AGENT_JUPITER:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
        xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
        xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="guest">
            <tns:read>
                <tns:include name="/home/user/public/**"/>
             </tns:read>
        </tns:sandbox>
```

```
    </tns:agent>
</tns:userSandboxes>
```

exemplo

Este exemplo mostra como permitir que qualquer usuário com o nome do usuário do MQMD `account` seguido por um único dígito, por exemplo, `account4`, conclua as ações a seguir:

- Transfira qualquer arquivo do diretório `/home/account` ou de qualquer um de seus subdiretórios, excluindo o diretório `/home/account/private` no sistema em que o agente AGENT_SATURN está em execução
- Transfira qualquer arquivo para o diretório `/home/account/output` ou qualquer um de seus subdiretórios no sistema em que o agente AGENT_SATURN está em execução
- Leia mensagens de filas no Gerenciador de Filas locais começando com o prefixo `ACCOUNT.` a menos que ele comece com `ACCOUNT.PRIVATE.` (ou seja, tenha `PRIVATE` no segundo nível).
- Transfira dados nas filas começando com o prefixo `ACCOUNT.OUTPUT.` em qualquer gerenciador de filas.

Para permitir que um usuário com o nome do usuário MQMD `account` conclua essas ações, inclua o elemento `<sandbox>` a seguir no arquivo `UserSandboxes.xml`, no diretório de configuração do AGENT_SATURN:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:userSandboxes
            xmlns:tns="http://wmqfte.ibm.com/UserSandboxes"
            xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://wmqfte.ibm.com/UserSandboxes UserSandboxes.xsd">
    <tns:agent>
        <tns:sandbox user="account[0-9]" userPattern="regex">
            <tns:read>
                <tns:include name="/home/account/**"/>
                <tns:include name="ACCOUNT.**" type="queue"/>
                <tns:exclude name="ACCOUNT.PRIVATE.**" type="queue"/>
                <tns:exclude name="/home/account/private/**"/>
                                                    </tns:read>
            <tns:write>
                <tns:include name="/home/account/output/**"/>
                <tns:include name="ACCOUNT.OUTPUT.**" type="queue"/>
            </tns:write>
        </tns:sandbox>
    </tns:agent>
</tns:userSandboxes>
```

Referências relacionadas

“Verificações adicionais para transferências curingas” na página 597
Se um agente tiver sido configurado com um ambiente de simulação do usuário ou do agente para restringir os locais para os quais e dos quais o agente pode transferir arquivos, é possível especificar se verificações adicionais devem ser feitas em transferências curingas para esse agente.

O arquivo MFT `agent.properties`

Verificações adicionais para transferências curingas

Se um agente tiver sido configurado com um ambiente de simulação do usuário ou do agente para restringir os locais para os quais e dos quais o agente pode transferir arquivos, é possível especificar se verificações adicionais devem ser feitas em transferências curingas para esse agente.

Propriedade additionalWildcardSandboxChecking

Para ativar a verificação adicional para transferências curingas, inclua a propriedade a seguir no arquivo `agent.properties` para o agente que você deseja verificar.

```
additionalWildcardSandboxChecking=true
```

Quando essa propriedade estiver configurada como true e o agente fizer uma solicitação de transferência que tente ler um local que estiver fora do ambiente de simulação definido para correspondência de arquivos do curinga, a transferência falhará. Se houver múltiplas transferências dentro de uma solicitação de transferência e uma dessas solicitações falhar devido à tentativa de ler um local fora do ambiente de simulação, a transferência inteira falhará. Se a verificação falhar, a razão para a falha será fornecida em uma mensagem de erro.

Se a propriedade adiciontionalWildcardSandboxChecking for omitida do arquivo `agent.properties` de um agente ou for configurada como false, nenhuma verificação adicional será feita em transferências curingas para esse agente.

Mensagens de erro para verificação de curinga

As mensagens que são relatadas quando uma solicitação de transferência curinga é feita em um local fora de um local do ambiente de simulação configurado são conforme a seguir.

A mensagem a seguir ocorrerá quando um caminho de arquivo curinga em uma solicitação de transferência estiver localizado fora do ambiente de simulação restrito:

```
BFGSS0077E: A tentativa de ler o caminho de arquivo: path foi negada.
O caminho do arquivo foi localizado fora do ambiente restrito de simulação de transferência.
```

A mensagem a seguir ocorrerá quando uma transferência dentro de uma solicitação de múltiplas transferências contiver uma solicitação de transferência curinga no local em que o caminho estiver localizado fora do ambiente de simulação restrito:

```
BFGSS0078E: A tentativa de ler o caminho de arquivo: path foi ignorada porque outro item de
transferência na transferência gerenciada tentou ler fora do ambiente restrito de simulação de
transferência.
```

A mensagem a seguir ocorrerá quando um arquivo estiver localizado fora do ambiente de simulação restrito:

```
BFGSS0079E: A tentativa de ler o arquivo file path foi negada.
O arquivo está localizado fora da sandbox de transferência restrita.
```

A mensagem a seguir ocorrerá em uma solicitação de múltiplas transferências na qual outra solicitação de transferência curinga fez essa ser ignorada:

```
BFGSS0080E: A tentativa de ler o arquivo: file path foi ignorada pois outro item de
transferência na transferência gerenciada tentou ler fora do ambiente restrito de simulação de
transferência.
```

No caso de transferências de arquivos simples que não incluem curingas, a mensagem que é relatada quando a transferência envolve um arquivo que está localizado fora do ambiente de simulação é inalterada desde as liberações anteriores:

```
Falha com BFGIO0056E: A tentativa de ler o arquivo "FILE" foi negada.
O arquivo está localizado fora da sandbox de transferência restrita.
```

Referências relacionadas

“Trabalhando com ambientes de simulação do usuário do MFT” na página 594
É possível restringir a área do sistema de arquivos à qual os arquivos podem ser transferidos para/de com base no nome de usuário MQMD do usuário que solicita a transferência.

“Trabalhando com ambientes de simulação do agente MFT” na página 593
Para incluir um nível de segurança adicional no Managed File Transfer, é possível restringir a área de um sistema de arquivos que um agente pode acessar.

O arquivo MFT `agent.properties`

Configurando a criptografia SSL ou TLS para o MFT

É possível usar SSL ou TLS que podem ser usados com o IBM MQ Managed File Transfer para proteger a comunicação entre os agentes e seus gerenciadores de fila de agentes, comandos e os gerenciadores de filas aos quais eles estão se conectando e os vários gerenciadores de filas para conexões do gerenciador de filas dentro de sua topologia

Antes de começar

É possível usar a criptografia SSL ou TLS para criptografar mensagens que estão fluindo por uma topologia do IBM MQ Managed File Transfer . Isso inclui:

- Mensagens que são transmitidas entre um agente e seu gerenciador de filas do agente
- Mensagens para comandos e gerenciadores de filas aos quais eles estão se conectando.
- Mensagens internas que fluem entre os gerenciadores de fila do agente, gerenciadores de fila de comandos e gerenciador de fila de coordenação dentro da topologia.

Sobre esta tarefa

Para obter informações gerais sobre como usar SSL com o IBM MQ, veja “Trabalhando com SSL/TLS” na página 280. Nos termos do IBM MQ, o Managed File Transfer é um aplicativo de cliente Java padrão.

Siga estas etapas para usar o SSL com o Managed File Transfer:

Procedimento

1. Crie um arquivo truststore e, opcionalmente, um arquivo keystore (estes arquivos podem ser o mesmo arquivo). Se você não precisar de uma autenticação de cliente (ou seja, SSLCAUTH=OPTIONAL em canais), não será necessário fornecer um keystore. Você precisa de um armazenamento confiável apenas para autenticar o certificado do gerenciador de fila

   O algoritmo de chave usado para criar certificados para o armazenamento confiável e keystores deve ser RSA para trabalhar com o IBM MQ.
2. Configure o gerenciador de filas do IBM MQ para usar o SSL.

   Para obter informações sobre configuração de um gerenciador de filas para usar SSL usando o IBM MQ Explorer, por exemplo, veja Configurando SSL em gerenciadores de filas.
3. Salve os arquivos truststore e keystore (se houver um) em um local apropriado. Um local sugerido é o diretório *config_directory*/*coordination_qmgr*/agents/*agent_name*.
4. Configure as propriedades de SSL conforme necessário para cada gerenciador de filas ativado para SSL no arquivo de propriedades do Managed File Transfer apropriado. Cada conjunto de propriedades faz referência a um gerenciador de filas separado (agente, coordenação e comando), embora um gerenciador de fila possa executar duas ou mais funções.

   Uma das propriedades **CipherSpec** ou **CipherSuite** é necessária, caso contrário, o cliente tentará se conectar sem o SSL. Ambas as propriedades, **CipherSpec** ou **CipherSuite**, são fornecidas devido às diferenças de terminologia entre o IBM MQ e o Java. O Managed File Transfer aceita uma das propriedades e faz a conversão necessária, para que você não precise configurar ambas as propriedades. Se você especificar as propriedades **CipherSpec** ou **CipherSuite**, **CipherSpec** terá precedência.

   A propriedade **PeerName** é opcional. É possível configurar a propriedade como o Nome Distinto do gerenciador de filas ao qual quer se conectar. O Managed File Transfer rejeita conexões com um servidor SSL incorreto com um Nome Distinto que não corresponde.

   Configure as propriedades **SslTrustStore** e **SslKeyStore** como nomes de arquivos que apontam para os arquivos truststore e keystore. Se estiver configurando essas propriedades para um agente já em execução, pare e reinicie o agente a fim de se reconectar no modo SSL.

   Os arquivos de propriedades contêm senhas de texto simples, assim considere configurar as permissões do sistema de arquivos apropriadas.

   Para obter informações adicionais sobre propriedades SSL, consulte “Propriedades SSL/TLS para MFT” na página 600.
5. Se um gerenciador de filas do agente usar SSL, não será possível fornecer os detalhes necessários ao criar o agente. Use as etapas a seguir para criar o agente:
   a) Crie o agente usando o comando **fteCreateAgent**. Você recebe um aviso sobre não ser possível publicar a existência do agente no gerenciador de filas de coordenação.

b) Edite o arquivo `agent.properties` que foi criado pela etapa anterior para incluir as informações de SSL. Quando o agente é iniciado com êxito, a publicação é tentada novamente.

6. Se agentes ou instâncias do IBM MQ Explorer estiverem em execução enquanto as propriedades de SSL no arquivo `agent.properties` ou arquivo no `coordination.properties` forem mudadas, o agente ou o IBM MQ Explorer deverá ser reiniciado.

Referências relacionadas

O arquivo MFT `agent.properties`

Propriedades SSL/TLS para MFT

Alguns arquivos de propriedades MFT incluem propriedades SSL e TLS. É possível usar SSL ou TLS com IBM MQ e Managed File Transfer para evitar conexões não autorizadas entre agentes e gerenciadores de filas, além de criptografar o tráfego de mensagens entre agentes e gerenciadores de filas.

Os arquivos de propriedades MFT a seguir incluem propriedades SSL:

- Propriedades SSL/TLS para o arquivo MFT `agent.properties`
- Propriedades SSL/TLS para o arquivo MFT `coordination.properties`
- Propriedades SSL/TLS para o arquivo MFT `command.properties`
- Propriedades SSL/TLS para o arquivo MFT `logger.properties`

Para obter informações sobre como usar SSL ou TLS com Managed File Transfer, consulte “Configurando a criptografia SSL ou TLS para o MFT” na página 598.

No IBM WebSphere MQ 7.5, é possível utilizar variáveis de ambiente em algumas propriedades Managed File Transfer que representam locais de arquivo ou de diretório. Isso permite que os locais de arquivos e diretórios usados ao executar partes do produto variem, dependendo das mudanças de ambiente, tais como qual usuário está executando o processo. Para obter mais informações, consulte O uso de variáveis de ambiente nas propriedades MFT.

Conceitos relacionados

Opções de configuração do MFT em Multiplataformas

Referências relacionadas

O uso de variáveis de ambiente nas propriedades MFT

Conectando-se a um gerenciador de filas no modo cliente com autenticação de canal

O IBM MQ usa registros de autenticação de canal para controlar mais precisamente o acesso em um nível de canal. Isso significa que, por padrão, os gerenciadores de filas recém-criados rejeitam conexões do cliente do componente Managed File Transfer .

Para obter mais informações sobre autenticação de canal, veja “Registros de Autenticação de Canal” na página 53.

Se a configuração de autenticação de canal para o SVRCONN usada pelo Managed File Transfer especifica um ID de MCAUSER não privilegiado, deve-se conceder registros de autoridade específicos para o gerenciador de filas, filas e tópicos, para permitir que o Managed File Transfer Agent e os comandos funcionem corretamente. Use o comando do MQSC SET CHLAUTH ou o comando do PCF Configurar Registro de Autenticação de Canal para criar, modificar ou remover registros de autenticação de canal. Para todos os agentes do Managed File Transfer que você deseja conectar ao gerenciador de filas do IBM MQ , é possível configurar um ID MCAUSER para usar para todos os seus agentes ou configurar um ID MCAUSER separado para cada agente.

Conceda a cada ID de MCAUSER as permissões a seguir:

- Registros de autoridade necessários para o gerenciador de filas:
  - connect
  - setid

  - inq
- Registros de autoridade necessários para filas.

  Para todas as filas específicas do agente, que são nomes de filas que terminam em *agent_name* na lista a seguir, deve-se criar esses registros de autoridade de fila para cada agente que você deseja conectar ao gerenciador de filas do IBM MQ usando uma conexão do cliente.

  - put, get, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)
  - put, get, setid, browse (SYSTEM.FTE.COMMAND.*agent_name*)
  - put, get (SYSTEM.FTE.DATA.*agent_name*)
  - put, get (SYSTEM.FTE.REPLY.*agent_name*)
  - put, get, inq, browse (SYSTEM.FTE.STATE.*agent_name*)
  - put, get, browse (SYSTEM.FTE.EVENT.*agent_name*)
  - put, get (SYSTEM.FTE)
- Registros de autoridade necessários para tópicos:
  - sub, pub (SYSTEM.FTE)
- Registros de autoridade necessários para transferências de arquivos.

  Se você tiver IDs de MCAUSER separados para o agente de origem e de destino, crie os registros de autoridade nas filas dos agentes na origem e no destino.

  Por exemplo, se o ID do MCAUSER do agente de origem for **user1** e o ID do MCAUSER do agente de destino for **user2**, configure as autoridades a seguir para os usuários do agente:

Usuário do AGENTE	Fila	Autoridade necessária
user1	SYSTEM.FTE.DATA.*destination_agent_name*	put
user1	SYSTEM.FTE.COMMAND.*destination_agent_name*	put
user2	SYSTEM.FTE.REPLY.*source_agent_name*	put
user2	SYSTEM.FTE.COMMAND.*source_agent_name*	put

Configurando SSL ou TLS entre o agente de ponte Connect:Direct e o nó Connect:Direct

Configure o agente ponte Connect:Direct e o nó Connect:Direct para conectar um ao outro por meio do protocolo SSL criando um keystore e um truststore configurando propriedades no arquivo de propriedades do agente ponte Connect:Direct.

Sobre esta tarefa

Estas etapas incluem instruções para obter suas chaves designadas por uma autoridade de certificação. Se você não usar uma autoridade de certificação, poderá gerar um certificado autoassinado. Para obter mais informações sobre como gerar um certificado autoassinado, veja “Trabalhando com SSL/TLS no AIX, Linux, and Windows” na página 298.

Estas etapas incluem instruções para criar um novo keystore e truststore para o agente ponte Connect:Direct. Se o agente de ponte do Connect:Direct já tiver um keystore e um armazenamento confiável que ele usa para conectar-se com segurança aos gerenciadores de filas do IBM MQ, será possível usar o keystore e o armazenamento confiável existentes quando conectar-se com segurança ao nó do Connect:Direct. Para obter mais informações, consulte “Configurando a criptografia SSL ou TLS para o MFT” na página 598.

Procedimento

Para o nó Connect:Direct, complete as seguintes etapas:

1. Gere um certificado assinado e chave para o nó Connect:Direct.

   É possível fazer isso usando a ferramenta de Gerenciamento de Chaves do IBM que é fornecida com o IBM MQ. Para obter informações adicionais, consulte “Trabalhando com SSL/TLS” na página 280.
2. Envie um pedido para uma autoridade de certificação para obter a chave assinada. Você recebe um certificado como resposta.
3. Crie um arquivo de texto; por exemplo, `/test/ssl/certs/CAcert`, que contenha a chave pública de sua autoridade de certificação.
4. Instale a opção Secure + no nó Connect:Direct .

   Se o nó já existe, você pode instalar a Opção Secure+ executando o instalador novamente, especificando o local da instalação existentee escolhendo instalar somente a Opção Secure+.
5. Crie um novo arquivo de texto; por exemplo, `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`.
6. Copie o certificado que você recebeu de sua autoridade de certificação e a chave privada, localizada no `/test/ssl/cd/privateKeys/`*`node_name`*`.key`, no arquivo de texto.

   O conteúdo de `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt` deve estar no formato a seguir:

```
-----BEGIN CERTIFICATE-----
MIICnzCCAgigAwIBAgIBGjANBgkqhkiG9w0BAQUFADBeMQswCQYDVQQGEwJHQjES
MBAGA1UECBMJSGFtcHNoaXJlMRAwDgYDVQQHEwdIdXJzbGV5MQwwCgYDVQQKEwNJ
Qk0xDjAMBgNVBAsTBU1RSVBUMQswCQYDVQQDEwJDQTAeFw0xMTAzMDExNjIwNDZa
Fw0yMTAyMjYxNjIwNDZaMFAxCzAJBgNVBAYTAkdCMRIwEAYDVQQIEwlIYW1wc2hp
cmUxDDAKBgNVBAoTA0lCTTEOMAwGA1UECxMFTVFGVEUxDzANBgNVBAMTBmJpbmJh
ZzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAvgP1QIklU9ypSKD1XoODo1yk
EyMFXBOUpZRrDVxjoSEC0vtWNcJ199e+Vc4UpNybDyBu+NkDlMNofX4QxeQcLAFj
WnhakqCiQ+JIAD5AurhnrwChe0MV3kjA84GKH/rOSVqtl984mu/lDyS819XcfSSn
cOOMsK1KbneVSCIV2XECAwEAAaN7MHkwCQYDVR0TBAIwADAsBglghkgBhvhCAQ0E
HxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0OBBYEFNXMIpSc
csBXUniW4A3UrZnCRsv3MB8GA1UdIwQYMBaAFDXY8rmj4lVz5+FVAoQb++cns+B4
MA0GCSqGSIb3DQEBBQUAA4GBAFc7klXa4pGKYgwchxKpE3ZF6FNwy4vBXS216/ja
8h/vl8+iv01OCL8t0ZOKSU95fyZLzOPKnCH7v+ItFSE3CIiEk9Dlz2U6WO9lICwn
l7PL72TdfaL3kabwHYVf17IVcuL+VZsZ3HjLggP2qHO9ZuJPspeT9+AxFVMLiaAb
8eHw
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,64A02DA15B6B6EF9

57kqxLOJ/gRUOIQ6hVK2YN13B4E1jAi1gSme0I5ZpEIG8CHXISKB7/0cke2FTqsV
lvI99QyCxsDWoMNt5fj51v7aPmVeS60bOm+UlGre8B/Ze18JVj2O4K2Uh72rDCXE
5e6eFxSdUM207sQDy20euBVELJtM2kOkL1ROdoQQSlU3XQNgJw/t3ZIx5hPXWEQT
rjRQO64BEhb+PzzxPF8uwzZ9IrUK9BJ/UUnqC6OdBR87IeA4pnJD1Jvb2ML7EN9Z
5Y+50hTKI8OGvBvWXO4fHyvIX5aslwhBoArXIS1AtNTrptPvoaP1zyIAeZ6OCVo/
SFo+A2UhmtEJeOJaZG2XZ3H495fAw/EHmjehzIACwukQ9nSIETgu4A1+CV64RJED
aYBCM8UjaAkbZDH5gn7+eBov0ssXAXWDyJBVhUOjXjvAj/e1h+kcSF1hax5D//AI
66nRMZzboSxNqkjcVd8wfDwP+bEjDzUaaarJTS7lIFeLLw7eJ8MNAkMGicDkycL0
EPBU9X5QnHKLKOfYHN/1WgUk8qt3UytFXXfzTXGF3EbsWbBupkT5e5+lYcX8OVZ6
sHFPNlHluCNy/riUcBy9iviVeodX8IomOchSyO5DKl8bwZNjYtUP+CtYHNFU5BaD
I+1uUOAeJ+wjQYKT1WaeIGZ3VxuNITJul8y5qDTXXfX7vxM5OoWXa6U5+AYuGUMg
/itPZmUmNrHjTk7ghT6i1IQOaBowXXKJBlMmq/6BQXN2IhkD9ys2qrvM1hdi5nAf
egmdiG50loLnBRqWbfR+DykpAhK4SaDi2F52Uxovw3Lhiw8dQP7lzQ==
-----END RSA PRIVATE KEY-----
```

7. Inicie o Secure+ Admin Tool.

   - Em sistemas AIX and Linux, execute o comando **spadmin.sh**.
   - Em sistemas Windows, clique em **Iniciar** > **Programas** > **Sterling Commerce Connect:Direct** > **CD Secure+ Admin Tool**

   O CD Secure+ Admin Tool inicia.
8. No CD Secure+ Admin Tool, dê um clique duplo na linha **.Local** para editar as configurações SSL ou TLS principais.

   a) Selecione **Ativar Protocolo SSL** ou **Ativar Protocolo TLS**, dependendo de qual protocolo você está usando.

   b) Selecione **Desativar Substituição**.

   c) Selecione pelo menos um Conjunto de códigos

d) Se desejar autenticação bidirecional, altere o valor de **Ativar Autenticação de Cliente** para Yes.

e) No campo **Certificado raiz confiável**, insira o caminho para o arquivo de certificado público de sua autoridade de certificação, `/test/ssl/certs/CAcert`.

f) No campo **Arquivo de certificado de chave**, insira o caminho para o arquivo que você criou, `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`.

9. Dê um clique duplo na linha **.Client** para editar as configurações SSL ou TLS principais.

a) Selecione **Ativar Protocolo SSL** ou **Ativar Protocolo TLS**, dependendo de qual protocolo você está usando.

b) Selecione **Desativar Substituição**.

Para o agente ponte Connect:Direct, execute as seguintes etapas:

10. Crie um truststore. É possível fazer isso criando uma chave simulada e então excluindo a chave simulada.

    É possível usar os seguintes comandos:

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. Importe o certificado público da autoridade de certificação dentro do truststore.

    É possível usar o seguinte comando:

```
keytool -import -trustcacerts -alias myCA
        -file /test/ssl/certs/CAcert
        -keystore /test/ssl/fte/stores/truststore.jks
```

12. Edite o arquivo de propriedades do agente ponte Connect:Direct.

    Inclua as seguintes linha em qualquer parte do arquivo:

```
cdNodeProtocol=protocol
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

    No exemplo nesta etapa, *protocol* é o protocolo que está usando, SSL ou TLSe *password* é a senha que especificou quando criou o armazenamento confiável.

13. Se deseja autenticação de duas vias, crie uma chave e certificado para o agente ponte Connect:Direct.

a) Crie um keystore e chave.

   É possível usar o seguinte comando:

```
keytool -genkey -keyalg RSA -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -validity 365
```

b) Gere uma solicitação de sinal.

   É possível usar o seguinte comando:

```
keytool -certreq -v -alias agent_name
        -keystore /test/ssl/fte/stores/keystore.jks -storepass password
        -file /test/ssl/fte/requests/agent_name.request
```

c) Importe o certificado que você recebe da etapa precedente no keystore. O certificado deve estar no formato x.509.

   É possível usar o seguinte comando:

```
keytool -import -keystore /test/ssl/fte/stores/keystore.jks
        -storepass password -file certificate_file_path
```

d) Edite o arquivo de propriedades do agente ponte Connect:Direct.

Inclua as seguintes linha em qualquer parte do arquivo:

```
cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
cdNodeKeystorePassword=password
```

No exemplo nesta etapa, *password* é a senha que especificou quando criou o keystore.

Tarefas relacionadas

Configurando a Ponte Connect:Direct

ALW Protegendo clientes AMQP

Você usa um intervalo de mecanismos de segurança para proteger conexões de clientes AMQP e assegurar que os dados sejam convenientemente protegidos na rede. É possível construir a segurança em seus aplicativos MQ Light. Também é possível usar os recursos de segurança existentes do IBM MQ com clientes AMQP, da mesma maneira que os recursos são usados para outros aplicativos.

Regras de autenticação de canal (CHLAUTH)

É possível usar as regras de autenticação de canal para restringir as conexões TCP para um gerenciador de filas. Os canais AMQP suportam o uso de regras de autenticação de canal configurados para seu gerenciador de filas. Se as regras de autenticação de canal forem definidas com um perfil que corresponda a quaisquer canais AMQP em seu gerenciador de filas, essas regras serão aplicadas a esses canais. Por padrão, a autenticação de canal é ativada em novos gerenciadores de filas do IBM MQ, portanto, deve-se concluir pelo menos alguma configuração antes de poder usar um canal AMQP.

Para obter mais informações sobre como configurar as regras de autenticação de canal para permitir conexões AMQP para o seu gerenciador de filas, consulte Criando e usando canais AMQP.

Autenticação de conexão (CONNAUTH)

É possível usar a autenticação de conexão para autenticar conexões com um gerenciador de filas. Os canais AMQP suportam o uso de autenticação de conexão para controlar o acesso ao gerenciador de filas a partir de aplicativos AMQP.

O protocolo AMQP usa a estrutura SASL (Camada de Segurança e Autenticação Simples) para especificar como uma conexão é autenticada. Há vários mecanismos de SASL e o IBM MQ suporta dois mecanismos de SASL : ANONYMOUS e PLAIN.

No caso de ANONYMOUS, nenhuma credencial é transmitida do cliente para o gerenciador de filas para autenticação. Se o objeto IBM MQ AUTHINFO que é especificado no atributo **CONNAUTH** do gerenciador de filas tiver um valor **CHCKCLNT** de REQUIRED ou REQDADM (se estiver conectando como um usuário administrativo), a conexão será recusada. Se o valor de **CHCKCLNT** for NONE ou OPTIONAL, a conexão será aceita.

No caso de PLAIN, uma senha e um nome do usuário são transmitidos do cliente para o gerenciador de filas para autenticação. Se o atributo IBM MQ AUTHINFO object que é especificado no gerenciador de filas **CONNAUTH** tiver um valor **CHCKCLNT** de NONE, a conexão será recusada. Se o valor de **CHCKCLNT** for OPTIONAL, REQUIREDou REQDADM (se estiver conectando como um usuário administrativo), o nome do usuário e a senha serão verificados pelo gerenciador de filas. O gerenciador de filas verifica o sistema operacional (se o objeto AUTHINFO for do tipo IDPWOS) ou um repositório LDAP (se o objeto AUTHINFO for do tipo IDPWLDAP)

A tabela a seguir resume esse comportamento de autenticação:

Tabela 101. Resumo dos mecanismos do SASL e da autenticação de conexão		
Mecanismo SASL	**Credenciais que foram transmitidas do cliente para o gerenciador de filas?**	**Valor CHKCLNT**
ANONYMOUS	No	REQUIRED ou REQDADM - conexão recusada NONE ou OPTIONAL - conexão aceita
PLAIN	Sim, nome do usuário e senha	REQUIRED, REQDADM ou OPTIONAL - nome do usuário e senha conferidos pelo gerenciador de filas NONE - conexão recusada

Se você estiver usando um cliente do MQ Light, será possível especificar as credenciais incluindo-as no endereço AMQP conectado, por exemplo:

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

Configurando MCAUSER em um canal

Os canais AMQP têm um atributo MCAUSER, que é possível ser usado para configurar o ID do usuário do IBM MQ que todas as conexões a esse canal estão autorizadas. Todas as conexões dos clientes AMQP a este canal adotam o ID MCAUSER, se você as tiver configurado. Esse ID do usuário é utilizado para autorização de mensagens em tópicos diferentes.

É recomendável usar a autenticação de canal (CHLAUTH) para proteger as conexões para os gerenciadores de filas. Se você estiver usando a autenticação de canal, será recomendável configurar o valor de MCAUSER para um usuário não privilegiado. Isso assegura que se uma conexão com um canal não for correspondida por uma regra CHLAUTH, a conexão não estará autorizada a executar nenhuma mensagem no gerenciador de filas.

Suporte SSL/TLS

Os canais AMQP suportam a criptografia de SSL/TLS usando as chaves a partir do repositório de chaves configurado para seu gerenciador de filas. As opções de configuração do canal AMQP para a criptografia de SSL/TLS suportam as mesmas opções que outros tipos de canal do MQ; é possível determinar uma especificação de cifra e se o gerenciador de filas requer certificados a partir de conexões do cliente AMQP.

Ao usar os atributos do FIPS do gerenciador de filas será possível controlar os conjuntos de cifras SSL/TLS, que será possível usar para proteger as conexões de clientes AMQP.

Para obter informações sobre como configurar um repositório de chaves para o gerenciador de fila, consulte “Trabalhando com SSL/TLS no AIX, Linux, and Windows” na página 298

Para obter informações sobre como configurar o suporte SSL/TLS para uma conexão do cliente AMQP, consulte Criando e usando canais AMQP.

V 9.4.0 V 9.4.0 No IBM MQ 9.4.0, o canal AMQP não suporta mais repositórios de chaves CMS no gerenciador de filas. É possível usar o comando **runmqakm** para converter um repositório de chaves CMS no formato PKCS #12 , que é suportado. Por exemplo, é possível usar o comando a seguir para

converter um repositório de chaves denominado `sslTest.kdb` do formato CMS para o formato PKCS #12 O novo repositório de chaves é denominado `sslTest.p12`, e protegido com a senha `passw0rd`

```
runmqakm -keydb -convert -type cms -db sslTest.kdb -stashed -new_format pkcs12 -target
sslTest.p12 -new_pw passw0rd
```

Java Serviço de Autenticação e Autorização (JAAS)

É possível, opcionalmente, configurar os canais AMQP com um módulo de login JAAS, que pode verificar o nome do usuário e a senha fornecidos por um cliente AMQP. Consulte “Configurando o JAAS para canais AMQP” na página 607.

Tarefas relacionadas

Desenvolvendo aplicativos clientes AMQP

Criando e utilizando canais AMQP

ALW Restringindo o controle do cliente AMQP

Quando uma conexão do cliente é feita AMQP que possui o mesmo identificador de cliente como uma conexão do cliente AMQP existente, a conexão do cliente existente está desconectado por padrão. Entretanto, será possível configurar o gerenciador de filas para restringir o comportamento de controle do cliente para que o controle seja possível apenas quando determinados critérios forem atendidos.

Por exemplo, desconectar a conexão de um cliente existente pode não ser apropriado se houver aplicativos AMQP em desenvolvimento por equipes diferentes que possam estar usando o mesmo identificador de cliente. Para abordar esse problema, você pode restringir o controle do cliente com base no nome do canal AMQP que está sendo utilizado, o endereço IP do cliente, e ID do Usuário do cliente (quando a autenticação de SASL é ativado).

Use as configurações dos atributos do gerenciador de filas **AdoptNewMCA** e **AdoptNewMCACheck** para especificar o nível necessário de restrição de controle do cliente, conforme detalhado na tabela a seguir:

*Tabela 102. Configurações **AdoptNewMCA** e **AdoptNewMCACheck** para restringir o controle do cliente*

AdoptNewMCA	AdoptNewMCACheck	Critérios verificados antes do controle de cliente ser permitido
NO ou indefinido	Não-aplicável	Nenhum. O controle do cliente é permitido para todas as conexões do cliente autenticadas e transmite todas as regras CHLAUTH.
TODOS (ou valor diferente de NO)	QM ou indefinido	Nenhum. O controle do cliente é permitido para todas as conexões do cliente autenticadas e transmite todas as regras CHLAUTH.
TODOS (ou valor diferente de NO)	Nome	ID do usuário (quando SASL ativado) Nome do canal
TODOS (ou valor diferente de NO)	ADDRESS	ID do usuário (quando SASL ativado) endereço IP

*Tabela 102. Configurações **AdoptNewMCA** e **AdoptNewMCACheck** para restringir o controle do cliente (continuação)*		
AdoptNewMCA	**AdoptNewMCACheck**	**Critérios verificados antes do controle de cliente ser permitido**
TODOS (ou valor diferente de NO)	ALL	ID do usuário (quando SASL ativado) Nome do canal endereço IP

Os atributos do gerenciador de filas **AdoptNewMCA** e **AdoptNewMCACheck** fazem parte da configuração do gerenciador de filas, que está definida na sub-rotina CANAIS. Nos sistemas IBM MQ for Windows e IBM MQ for Linux x86-64, modifique as informações de configuração usando o IBM MQ Explorer. Em outros sistemas, modifique as informações editando o arquivo de configuração `qm.ini` . Para obter informações sobre como modificar as informações de canais do gerenciador de filas, consulte Atributos de canais.

Tarefas relacionadas

Desenvolvendo aplicativos clientes AMQP

Criando e utilizando canais AMQP

ALW

Configurando o JAAS para canais AMQP

Java Os módulos customizados do Authentication and Authorization Service (JAAS) podem ser usados para autenticar credenciais de nome do usuário e senha transmitidas para um canal AMQP por um cliente AMQP quando ele se conecta.

Sobre esta tarefa

Você pode desejar usar um módulo JAAS customizado se já usar módulos JAAS para autenticação em outros sistemas baseados em Javae desejar reutilizar esses módulos para autenticar conexões AMQP para o MQ Como alternativa, você poderá desejar gravar um módulo JAAS customizado se os recursos de autenticação construídos no MQ não suportarem o mecanismo de autenticação que você deseja usar.

A configuração de módulos JAAS para canais AMQP é feita em um nível do gerenciador de filas. Isso significa que, se você configurar um módulo JAAS para autenticar conexões AMQP para o gerenciador de filas, esse módulo se aplicará a todos os canais AMQP. O nome do canal que chamou o módulo JAAS é passado para o módulo, permitindo codificar um log JAAS diferente em comportamento para diferentes canais.

Outras informações também são passadas para o módulo JAAS:

- O identificador de cliente do cliente AMQP que está tentando autenticar.
- O endereço de rede do cliente AMQP.
- O nome do canal que chamou o módulo JAAS.

Procedimento

Defina um módulo de configuração JAAS para canais AMQP concluindo as etapas a seguir:

1. Defina um arquivo `jaas.config` contendo uma ou mais sub-rotinas de configuração do módulo JAAS. A sub-rotina deve especificar o nome completo da classe Java que implementa a interface JAAS `javax.security.auth.spi.LoginModule` ..
   - Um arquivo padrão `jaas.config` é enviado com o produto e está localizado em *QM_data_directory*`/amqp/jaas.config`.

- Uma sub-rotina pré-configurada denominada MQXRConfig já está definida no arquivo padrão jaas.config.

2. Especifique o nome da sub-rotina a usar para canais AMQP.
   - Linux AIX Inclua uma propriedade no arquivo amqp_unix.properties.
   - Windows Inclua uma propriedade no arquivo amqp_win.properties.

   A propriedade tem a forma a seguir:

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   Por exemplo:

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```

3. Configure o ambiente do gerenciador de filas para incluir a classe do módulo customizado. O serviço AMQP deve ter acesso à classe Java configurada na sub-rotina de configuração JAAS .

   Você faz isso incluindo o caminho para a classe JAAS para o arquivo MQ service.env. Edite o arquivo service.env no diretório de configuração MQ (*MQ_config_directory*) ou o diretório de configuração do Gerenciador de Filas (*QM_config_directory*) para configurar a variável CLASSPATH para a localização da classe de módulo JAAS.

Como proceder a seguir

Um módulo de login de amostra JAAS é enviado com o produto no diretório *mq_installation_directory*/amqp/samples. O módulo de login do JAAS de amostra autentica todas as conexões do cliente, independentemente do nome do usuário ou senha com o qual o cliente se conecta.

É possível modificar o código-fonte da amostra e recompilá-lo para tentar a autenticação de somente usuários específicos com uma senha particular. Para configurar o canal AMQP em um sistema UNIX para usar o módulo de login do JAAS de amostra enviado com o produto:

1. Edite o arquivo /var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties e configure a propriedade com.ibm.mq.MQXR.JAASConfig=MQXRConfig.
2. Edite o arquivo /var/mqm/service.env e configure a propriedade CLASSPATH=*mq_installation_location*/amqp/samples

O arquivo jaas.config já contém uma sub-rotina nomeada MQXRConfig que especifica a classe de amostra samples.JAASLoginModule como a classe do módulo de login. Nenhuma mudança é necessária para jaas.config antes de você tentar o módulo de amostra.

Tarefas relacionadas

Desenvolvendo aplicativos clientes AMQP

Criando e utilizando canais AMQP

Advanced Message Security

O Advanced Message Security (AMS) é um componente do IBM MQ que fornece um alto nível de proteção para dados sensíveis que fluem por meio da rede do IBM MQ, ao mesmo tempo em que não impacta os aplicativos finais.

Visão geral do Advanced Message Security

Os aplicativos do IBM MQ podem usar Advanced Message Security para enviar dados confidenciais, como transações financeiras de valor alto e informações pessoais, com níveis diferentes de proteção usando um modelo de criptografia de chave pública.

Conceitos relacionados

“Intercepção do Agente do Canal de Mensagens (MCA) e AMS” na página 660

A intercepção de MCA permite que um gerenciador de filas em execução sob o IBM MQ ative seletivamente as políticas a serem aplicadas para os canais de conexão do servidor.

Referências relacionadas

Códigos de retorno do GSKit usados em mensagens do AMS

Recursos e Funções do Advanced Message Security

O Advanced Message Security expande serviços de segurança do IBM MQ para fornecer dados de assinatura e criptografia no nível de mensagem. Os serviços expandidos garantem que os dados da mensagem não foram modificados entre quando eles foram originalmente colocados em uma fila e quando são recuperados. Além disso, o AMS verifica se um emissor de dados da mensagem está autorizado a colocar mensagens assinadas em uma fila de destino.

O AMS oferece as seguintes funções:

- Protege transações sensíveis ou de valor alto processadas pelo IBM MQ.
- Detecta e remove mensagens prejudiciais ou não autorizadas antes de serem processadas por uma aplicação de recepção.
- Verifica se as mensagens não foram modificadas durante a passagem de uma fila para outra.
- Protege os dados não só à medida que eles fluem através da rede, mas também quando são colocados em uma fila.
- Assegura as aplicações proprietárias e gravadas pelo cliente para IBM MQ.
- z/OS No IBM MQ 9.1.3, o IBM MQ for z/OS fornece a capacidade de, opcionalmente, remover e incluir a proteção do AMS de ou para mensagens que fluem pela rede, respectivamente. Isso é conhecido como *Intercepção do Agente do canal de mensagens (MCA) de servidor para servidor*.
- ALW Desde IBM MQ 9.1.4 e IBM MQ 9.1.0 Fix Pack 4, uma verificação é incluída no código de biblioteca do IBM MQ que é executado dentro do programa de aplicativo cliente. A verificação é executada antecipadamente em sua inicialização para ler o valor da variável de ambiente *AMQ_AMS_FIPS_OFF* e, se ele for configurado para qualquer valor, o código IBM Global Security Kit (GSKit) será executado no modo não FIPS nesse aplicativo.

Qualidades de proteção disponíveis com AMS

Existem três qualidades de proteção para Advanced Message Security, `Integrity`, `Privacy` e `Confidentiality`.

A proteção `Integrity` é fornecida por assinatura digital, que garante quem criou a mensagem e que a mensagem não foi alterada ou adulterada.

A proteção `Privacy` é fornecida por uma combinação de assinatura digital e criptografia. A criptografia assegura que os dados da mensagem estejam visíveis somente ao destinatário-alvo ou destinatários. Mesmo que destinatários desautorizados obtenham uma cópia dos dados da mensagem criptografada, eles não poderão visualizar os dados da mensagem real em si.

A proteção `Confidentiality` é fornecida por criptografia apenas com reutilização de chave opcional.

Efeito sobre o Desempenho

O AMS usa uma combinação de rotinas criptográficas simétricas e assimétricas para fornecer a assinatura digital e a criptografia. Como as operações de chave simétrica são muito rápidas em comparação com operações de chave assimétrica, que são de CPU intensiva, isso pode ter um impacto significativo nos custos da proteção de grandes números de mensagens com o AMS.

Rotinas criptográficas assimétricas

Por exemplo, ao colocar uma mensagem assinada, o hash de mensagem é assinado usando uma operação de chave assimétrica.

Ao colocar uma mensagem assinada, uma operação de chave assimétrica adicional é usada para verificar o hash assinado.

Portanto, um mínimo de duas operações de chave assimétrica é necessário por mensagem para assinar e verificar os dados da mensagem.

Rotinas criptográficas assimétricas e simétricas

Ao colocar uma mensagem criptografada, uma chave simétrica é gerada e depois criptografada usando uma operação de chave assimétrica para cada destinatário-alvo da mensagem.

Os dados da mensagem são criptografados com a chave simétrica. Ao colocar a mensagem criptografada, o destinatário-alvo precisa usar uma operação de chave assimétrica para descobrir a chave simétrica em uso para a mensagem.

Todas as três qualidades de proteção, portanto, contêm vários elementos de operações de chave assimétrica de CPU intensiva, que impactarão significativamente a taxa máxima do sistema de mensagens acessível para aplicativos que estejam colocando e obtendo mensagens.

As políticas `Confidentiality`, no entanto, permitem a reutilização de chave simétrica em uma sequência de mensagens. Economias significativas de custo de CPU podem ser feitas com políticas `Confidentiality` por meio da reutilização de chave simétrica. Esse modo de operação continua usando o formato PKCS#7 para compartilhar uma chave de criptografia simétrica. Entretanto, não há assinatura digital, o que elimina algumas das operações de chave assimétrica por mensagem. A chave simétrica ainda precisa ser criptografada com operações de chave assimétrica para cada destinatário, mas a chave simétrica pode ser reutilizada opcionalmente em diversas mensagens destinada aos mesmos destinatários. Se a reutilização de chave for permitida pela política, somente a primeira mensagem requererá operações de chave assimétrica. As mensagens subsequentes só precisam usar operações de chave simétrica.

Reutilização da chave

Com políticas do `Confidentiality` , é possível usar a abordagem de reutilização de chave simétrica para reduzir significativamente os custos envolvidos na criptografia de várias mensagens que são colocadas na mesma fila e destinadas ao mesmo destinatário ou destinatários.

Por exemplo, ao colocar 10 mensagens criptografadas no mesmo conjunto de destinatários, uma chave simétrica é gerada e, em seguida, criptografada para a primeira mensagem, usando uma operação de chave assimétrica para cada destinatário-alvo da mensagem.

Com base nos limites controlados de políticas, a chave simétrica criptografada pode ser reutilizada pelas mensagens subsequentes destinadas para os mesmos destinatários. Para permitir que a chave simétrica seja reutilizada por mensagens subsequentes, o aplicativo deve manter a fila aberta após colocar uma mensagem na fila. A chave simétrica não pode ser reutilizada por operações MQPUT1. Um aplicativo que está obtendo mensagens criptografadas pode aplicar a mesma otimização, no sentido de que o aplicativo pode detectar quando uma chave simétrica não foi mudada e evitar a despesa de recuperá-la.

Neste exemplo, 90% das operações de chave assimétrica podem ser evitadas tanto colocando quanto obtendo aplicativos pela reutilização da mesma chave.

Para obter mais informações sobre como usar a reutilização de chave, consulte:

- Comando MQSC SET POLICY
- Comando de controle setmqspl
- IBM i Comando IBM i SETMQMSPL

Conceitos chave no AMS

Aprenda sobre os conceitos chave no Advanced Message Security para entender como a ferramenta funciona e como gerenciar de forma efetiva.

Infraestrutura de chave pública e Advanced Message Security

A infraestrutura de chave pública (PKI) é um sistema de recursos, políticas e serviços que suportam o uso de criptografia de chave pública para obter comunicação segura.

Não há um padrão que defina os componentes de uma infraestrutura de chave pública, mas um PKI geralmente envolve o uso de certificados de chave pública e inclui autoridades de certificação (CA) e outras autoridades de registro (RA) que fornecem os serviços a seguir:

- Emissão de certificados digitais
- Validação de certificados digitais
- Revogação de certificados digitais
- Distribuindo Certificados

A identidade de usuários e aplicativos é representada pelo campo **nome distinto (DN)** em um certificado associado às mensagens assinadas ou criptografadas. O Advanced Message Security usa essa identidade para representar um usuário ou um aplicativo Para autenticar essa identidade, o usuário ou aplicativo deve ter acesso ao armazenamento de chaves no qual o certificado e a chave privada associada são armazenados. Cada certificado é representado por um rótulo no keystore.

Conceitos relacionados

“Usando keystores e certificados com o AMS” na página 654
Para fornecer proteção criptográfica transparente para aplicativos IBM MQ, o Advanced Message Security usa o arquivo keystore, em que certificados de chave pública e uma chave privada são armazenados. Ativado z/OS, um conjunto de chaves SAF é usado em vez de um arquivo keystore

Certificados digitais no AMS

O Advanced Message Security associa usuários e aplicativos com certificados digitais padrão X.509. Certificados X.509 são geralmente assinados por uma autoridade de certificação (CA) confiável e envolvem as chaves públicas e privadas que são usadas para criptografia e decriptografia.

Os certificados digitais fornecem proteção contra identidades pela ligação de uma chave pública a seu proprietário, se esse for um indivíduo, um gerenciador de filas ou alguma outra entidade. Certificados digitais também são conhecidos como certificados de chave pública, porque eles oferecem a garantia sobre a propriedade de uma chave pública quando você usa um esquema de chave assimétrica. Este esquema requer que uma chave pública e uma chave privada sejam geradas para um aplicativo. Os dados criptografados com a chave pública só podem ser decriptografados usando a chave privada correspondente enquanto os dados criptografados com a chave privada só podem ser decriptografados usando a chave pública correspondente. A chave privada é armazenada em um arquivo do banco de dados de chaves protegido por senha. Apenas o proprietário tem acesso à chave privada usada para decriptografar mensagens que foram criptografadas usando a chave pública correspondente.

Se chaves públicas forem enviadas diretamente por seu proprietário a outra entidade, há um risco de que a mensagem possa ser interceptada e a chave pública ser substituída por outra. Isso é conhecido como um ataque "man-in-the-middle". A solução é trocar chaves públicas por meio de terceiros confiáveis, fornecendo ao usuário uma garantia segura de que a chave pública pertence à entidade com a qual você está se comunicando. Em vez de enviar sua chave pública diretamente, peça aos terceiros confiáveis para incorporá-la a um certificado digital. Os terceiros confiáveis que emitem certificados digitais são chamados de autoridade de certificação (CA).

Para obter mais informações sobre certificados digitais, consulte O que é um certificado digital.

Um certificado digital contém a chave pública de uma entidade e afirma que a chave pública pertence àquela entidade:

- quando um certificado for para uma entidade individual, ele será chamado de *certificado pessoal* ou *certificado de usuário*.
- quando um certificado for para uma autoridade de certificação, ele será chamado de *certificado CA* ou *certificado de assinante*.

Nota: O Advanced Message Security suporta certificados autoassinados em ambos o Java e aplicativos nativos

Conceitos relacionados

"Criptografia" na página 11

Criptografia é o processo de conversão entre texto legível, chamado *texto simples*, e uma forma ilegível, chamada *texto cifrado*.

Multi *Gerenciador de autoridade de objeto e AMS*

Em multiplataformas, o gerenciador de autoridade de objeto (OAM) é o componente de serviço de autorização fornecido com os produtos IBM MQ.

O acesso às entidades Advanced Message Security é controlado por meio de grupos de IBM MQ do usuário e do OAM. Os administradores podem usar a interface da linha de comandos para conceder ou revogar as autorizações, conforme necessário. Diferentes grupos de usuários podem ter diferentes tipos de autoridade de acesso para os mesmos objetos. Por exemplo, um grupo pode executar ambas as operações PUT e GET para uma fila específica enquanto outro grupo pode ter permissão somente para navegar na fila. Da mesma forma, alguns grupos podem ter a autoridade GET e PUT a para uma fila, mas não têm permissão para alterar ou excluir a fila.

Através do OAM, é possível controlar:

- Acesso a objetos Advanced Message Security por meio da Message Queue Interface (MQI). Quando um programa de aplicativo tenta acessar objetos, o OAM verifica se o perfil do usuário fazendo a solicitação tem a autorização para a operação solicitada. Isso significa que as filas e as mensagens nas filas podem ser protegidas contra acesso não autorizado.
- Permissão de usar comandos PCF e MQSC.

Conceitos relacionados

Gerenciador de autoridade de objeto

Visão geral da Message Queue Interface

Tecnologia suportada pelo Advanced Message Security

O Advanced Message Security depende de vários componentes de tecnologia para fornecer uma infraestrutura de segurança.

O Advanced Message Security suporta as interfaces de programação de aplicativos (APIs) do IBM MQ a seguir:

- MQI (Message Queue Interface)
- IBM MQ Java Message Service (JMS) 1.0.2 e 1.1.
- IBM MQ Classes base para o Java
- Classes do IBM MQ para .Net em um modo não gerenciado

Nota: O Advanced Message Security suporta autoridades de certificado compatíveis com X.509.

Limitações conhecidas de AMS

Há várias opções do IBM MQ que não são suportadas ou têm limitações para o Advanced Message Security.

- As opções do IBM MQ a seguir não são suportadas ou têm limitações:

  Publicação/assinatura

  Um dos principais benefícios de um modelo de sistema de mensagens de publicação/assinatura sobre ponto a ponto é que os aplicativos de envio e de recebimento não precisam saber nada um do outro para que os dados sejam enviados e recebidos. Esse benefício é negado pelo uso de políticas do Advanced Message Security que devem definir destinatários-alvo ou assinantes autorizados. É possível para um aplicativo publicar em um tópico por meio de uma definição de fila de alias que é protegida por uma política, também é possível para um aplicativo de assinatura obter mensagens de uma fila protegida por política. Não é possível designar uma política diretamente a uma sequência de tópicos, as políticas podem ser designadas somente a definições de filas.

Conversão de dados do canal
A carga útil protegida de uma mensagem protegida Advanced Message Security é transmitida usando formato binário, isso assegura que a conversão de dados em um canal entre aplicativos não invalide o trecho da mensagem. Os aplicativos que recuperam mensagens de uma fila protegida por política devem solicitar a conversão de dados, a conversão da carga útil protegida será tentada após as mensagens terem sido verificadas e desprotegidas com sucesso.

Listas de distribuição
As políticas Advanced Message Security podem ser usadas ao proteger os aplicativos que colocam mensagens em listas de distribuição, desde que cada fila de destino na lista tenha uma política idêntica definida. Se políticas inconsistentes forem identificadas quando um aplicativo abrir uma lista de distribuição, a operação aberta falhará e um erro de segurança retornado para o aplicativo.

Segmentação da mensagem do aplicativo
O tamanho das mensagens protegidas por política aumentará e não será possível para os aplicativos especificarem com precisão os limites de segmento de uma mensagem.

Aplicativos que usam o IBM MQ classes for .NET em um modo gerenciado (conexões do cliente)
Os aplicativos que usam o IBM MQ classes for .NET em um modo gerenciado (conexões do cliente) não são suportados.

Nota: A intercepção de MCA pode ser usada para permitir que clientes não suportados usem o AMS.

O cliente do serviço de mensagens para aplicativos .NET (XMS) em um modo gerenciado
O cliente do serviço de mensagens para aplicativos .NET (XMS) em um modo gerenciado não é suportado.

Nota: A intercepção de MCA pode ser usada para permitir que clientes não suportados usem AMS.

Filas do IBM MQ processadas pela ponte do IMS
As filas do IBM MQ processadas pela ponte do IMS não são suportadas.

Nota: O AMS é suportado nas filas de ponte do CICS. É necessário usar o mesmo ID do usuário para MQPUT (criptografia) e MQGET (decriptografia) nas filas de ponte do CICS.

Colocar em espera de getter
Colocar em getter getter não é suportado para aplicativos getter com relação a filas que possuem políticas AMS definidas para eles.

z/OS **Intercepção do servidor para servidor MCA**
No IBM MQ for z/OS 9.1.3, a interceptação MCA de servidor para servidor é compatível apenas com os tipos de canal emissor, servidor, receptor e solicitante.

- Os usuários devem evitar colocar mais de um certificado com o mesmo Nome distinto em um único arquivo keystore, porque a opção de qual certificado usar ao proteger uma mensagem é indefinida.
- O AMS não será suportado no JMS se a propriedade **WMQ_PROVIDER_VERSION** estiver configurada como 6.
- O interceptor do AMS não é suportado para canais AMQP ou MQTT.

z/OS Advanced Message Security interception on message channels

On z/OS, Advanced Message Security (AMS) interception provides an additional option of security policy protection (SPLPROT) to sender, server, receiver, and requester channels, allowing you to support AMS and to communicate with business partners who do not support AMS.

Taking the example of a clearing house communicating with a bank, Figure 1 shows that, without AMS interception, both sides of the system need to support AMS.

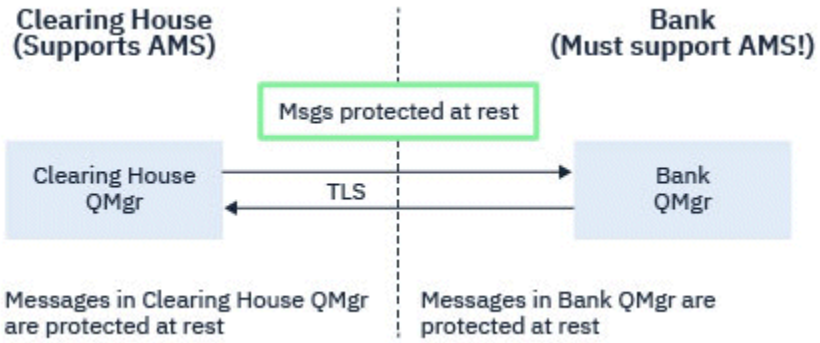


Figure 32. Usage of AMS without AMS interception

A key benefit of the AMS interception option is, that if your enterprise has AMS configured, and not all of your business partners support AMS, you can remove protection from outbound messages and protect inbound messages on channels to and from those business partners that do not support AMS.

Using the example of a clearing house and banks, this scenario is shown in Figure 2, where there is a message flow between the clearing house, banks, and business partners where some institutions have AMS, and others do not.

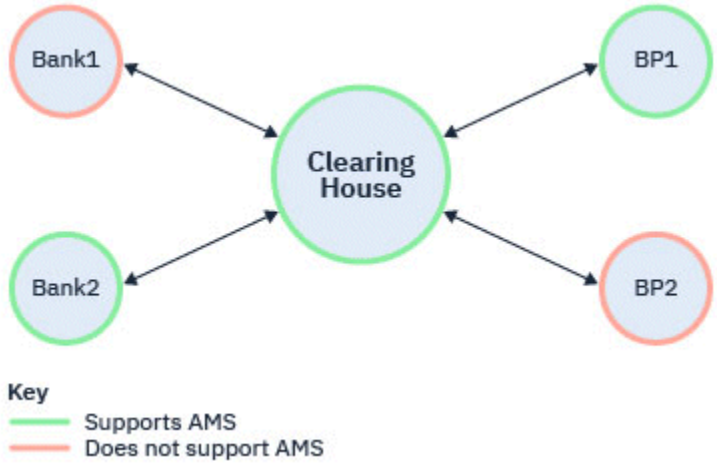


Figure 33. Some partners support AMS and some do not

Typically the channels are TLS enabled.

However, there might be a case where some banks and business partners do not support AMS, and there is a requirement to be able to exchange messages between all banks and business partners. This scenario is shown in Figure 3

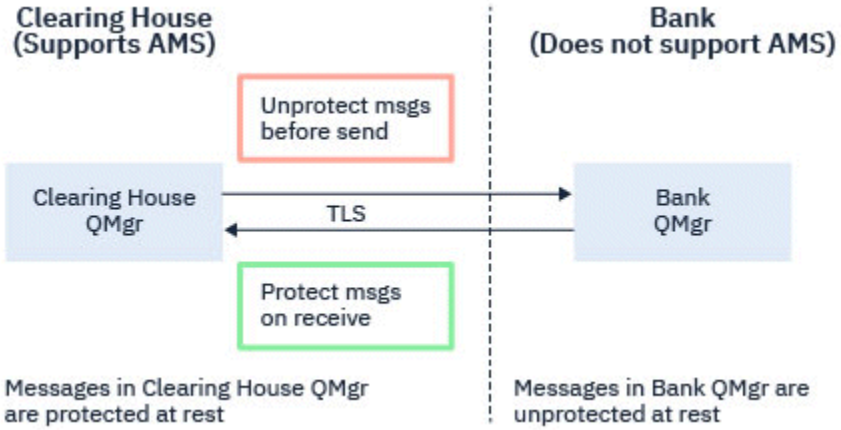


Figure 34. Message flow between business partners

Related tasks

Server-to-server message channel interception example configurations

z/OS *AMS interception on server-to-server message channels*

Server-to-server message channel interception provides a means to control if messages should have any applicable Advanced Message Security (AMS) policies applied to them, when sender type message channel agents get messages from transmission queues, and receiver type message channel agents put messages to target queues.

This allows AMS protection to be enabled on a queue manager when communicating, using server-to-server message channels of type sender, server, receiver, and requester, with a queue manager that does not have AMS enabled.

That is, AMS protected messages in AMS enabled queue managers can be unprotected prior to being sent to non-AMS enabled queue managers, and unprotected messages received from non-AMS enabled queue managers can be protected, by applicable AMS policies, on AMS enabled queue managers.

Configuring server-to-server message channel interception

Server-to-server message channel interception is configured with the SPLPROT attribute on channels with a channel type of sender, server, receiver, or requester. The available options to configure the behavior are dependent on the channel type specified:

PASSTHRU
: Transmita, de maneira inalterada, quaisquer mensagens enviadas ou recebidas pelo agente do canal de mensagens para esse canal.

  Este valor é válido para canais com um tipo de canal (**CHLTYPE**) de SDR, SVR, RCVR ou RQSTR, e é o valor padrão.

REMOVE
: Remova qualquer proteção do AMS das mensagens recuperadas da fila de transmissão pelo agente do canal de mensagens e envie as mensagens para o parceiro.

  Quando o agente do canal de mensagens recebe uma mensagem da fila de transmissão, se uma política do AMS for definida para a fila de transmissão, ela será aplicada para remover qualquer proteção do AMS da mensagem antes de enviar a mensagem pelo canal. Se uma política do AMS não estiver definida para a fila de transmissão, a mensagem será enviada no estado em que se encontra.

  Esse valor é válido apenas para canais com um tipo de canal de SDR ou SVR.

ASPOLICY
: Com base na política definida para a fila de destino, aplique a proteção do AMS nas mensagens de entrada antes de colocá-las na fila de destino.

Quando o agente do canal de mensagens recebe uma mensagem de entrada, se uma política do AMS estiver definida para a fila de destino, a proteção do AMS será aplicada à mensagem antes de a mensagem ser colocada na fila de destino. Se uma política do AMS não estiver definida para a fila de destino, a mensagem será colocada na fila de destino no estado em que se encontra.

Esse valor é válido apenas para canais com um tipo de canal de RCVR ou RQSTR.

User ID for message channel interception

The requirement for user IDs used with server-to-server message channel interception are the same as those for existing AMS enabled applications. For a running channel, the sending message channel agent gets messages from a transmission queue and the receiving message channel agent puts messages to target queues. The message channel agent user ID (MCAUSER) field, set on server to server channels, defines the user ID under which message channel agents perform put and get requests.

With server-to-server message channel interception, AMS functions are performed during get and put requests, as with other AMS enabled applications. Therefore, message channel agent user Ids have the same requirements as those for AMS application user IDs.

The MCAUSER used to perform the put and get is configurable, and dependent on whether it is an outbound or inbound channel. See MCAUSER for details of how the chosen user ID performs actions on the message channel agent. As such, the user ID that the channel initiator is running under is the user ID that is to be used for AMS functions performed during server-to-server message channel interception. Therefore, these user IDs have the same requirements as those for AMS application user IDs.

Authentication is performed using the existing rules for the channel detailed for channels with PUTAUT configuration. See user IDs used by the channel initiator for more information.

Note: Server-to-server message channel interception does not take into account the value of the PUTAUT channel attribute.

Message size and MAXMSGL

Due to AMS protection, the message size of protected messages will be larger than the original message size.

Protected messages are larger than unprotected messages. Therefore, the value of the **MAXMSGL** attribute, on both queues and channels, might need to be altered to take into account the size of protected messages.

Related reference

Server-to-server message channel interception example configurations

Manipulação de erros para o AMS

O IBM MQ Advanced Message Security define uma fila de manipulação de erros para gerenciar mensagens que contêm erros ou mensagens que não podem ser desprotegidas.

As mensagens defeituosas são tratadas como casos excepcionais. Se uma mensagem recebida não atender aos requisitos de segurança para a fila em que ela está, por exemplo, se a mensagem estiver assinada quando deveria estar criptografada ou a decriptografia ou verificação de assinatura falhar, a mensagem será enviada para a fila de manipulação de erros. Uma mensagem pode ser enviada para a fila de manipulação de erros pelas razões a seguir:

- Incompatibilidade de quality of protection - uma incompatibilidade de quality of protection (QOP) existe entre a mensagem recebida e a definição de QOP na política de segurança.
- Erro de decriptografia - a mensagem não pode ser decriptografada.
- Erro de cabeçalho PDMQ - o cabeçalho da mensagem do Advanced Message Security (AMS) não pode ser acessado.
- Incompatibilidade de tamanho - comprimento de uma mensagem após a decriptografia ser diferente daquela esperada.

- Incompatibilidade de intensidade de algoritmo de criptografia - o algoritmo de criptografia de mensagem é mais fraco do que o necessário.
- Erro desconhecido - ocorreu um erro inesperado.

AMS usa o sistema SYSTEM.PROTECTION.ERROR.QUEUE como sua fila de manipulação de erros.. Todas as mensagens colocadas pelo IBM MQ AMS no sistema SYSTEM.PROTECTION.ERROR.QUEUE são precedidos por um cabeçalho MQDLH..

O administrador do IBM MQ também pode definir o SYSTEM SYSTEM.PROTECTION.ERROR.QUEUE como uma fila de alias apontando para outra fila.

z/OS No IBM MQ for z/OS, se a intercepção do Message Channel Agent (MCA) do servidor para o servidor estiver em uso:

- Se, por um dos motivos mencionados anteriormente, o IBM MQ AMS mover mensagens da fila de transmissão para a fila de manipulação de erros, o MCA do remetente simplesmente continuará processando a próxima mensagem disponível na fila de transmissão.
- Em geral, as regras de canal existentes se aplicam a:
  - colocar mensagens na Fila de mensagens não entregues e
  - as ações executadas se inclusões na Fila de mensagens não entregues falharem.

Consulte “Mensagens não entregues para AMS on z/OS” na página 617 para obter informações adicionais sobre cenários específicos.

z/OS *Mensagens não entregues para AMS on z/OS*

Cenários específicos relacionados à intercepção do Agente do canal de mensagens de servidor para servidor no IBM MQ for z/OS.

No IBM MQ for z/OS, se a intercepção do Message Channel Agent (MCA) do servidor para o servidor estiver em uso:

- Se, após ter desprotegido e desprotegido uma mensagem, o MCA do emissor não entregar uma mensagem por algum motivo, por exemplo, porque a mensagem é muito grande para o canal, se o atributo do canal emissor USEDLQ estiver configurado como YES, o MCA do emissor moverá a mensagem para a Fila de Devoluções (DLQ) local.

  Se o SYSTEM.DEAD.LETTER.QUEUE estiver sendo usado como a DLQ local, a mensagem será colocada desprotegida.

  Nota: O IBM MQ AMS não suporta a proteção de mensagens colocadas em filas do sistema.

  Se um DLQ nomeado estiver sendo usado como o DLQ local, a mensagem será colocada como protegida, se você tiver definido uma política do IBM MQ AMS com o mesmo nome que o DLQ nomeado, e desprotegida, se você não definiu uma política adequada.
- Se uma mensagem não puder ser colocada na DLQ local por algum motivo e se o NPMSPEED do canal estiver configurado como NORMAL, ou a mensagem for uma mensagem persistente, o lote atual de mensagens será restaurado e o canal será colocado no estado RETRY. Caso contrário, a mensagem será descartada e o MCA do emissor continuará processando a próxima mensagem na fila de transmissão.
- Como as políticas de segurança não têm efeito sobre a SYSTEM.DEAD.LETTER.QUEUE ou as outras filas SYSTEM listadas em “Proteção da fila do sistema no AMS” na página 691, se o SYSTEM.DEAD.LETTER.QUEUE estiver em uso, as mensagens colocadas nesta fila por MCAs serão colocadas como estão. Ou seja, se as mensagens foram protegidas anteriormente, elas serão protegidas; caso contrário, elas serão colocadas desprotegidas.

  Se o atributo DEADQ do gerenciador de filas tiver sido configurado para o nome de uma fila de mensagens não entregues alternativa (não do sistema) e uma política do AMS com o mesmo nome não existir, as mensagens colocadas nessa fila pelos MCAs serão colocadas no estado em que se encontram. Ou seja, se as mensagens foram protegidas anteriormente, elas serão protegidas; caso contrário, elas serão colocadas desprotegidas.

Se o atributo DEADQ do gerenciador de filas tiver sido configurado para o nome de uma fila de mensagens não entregues alternativa (não sistema) e uma política do AMS com o mesmo nome que a DLQ existir, a política será usada para proteger mensagens colocadas nessa fila por MCAs. Se a mensagem já tiver sido protegida anteriormente, ela não será protegida novamente; isso é para evitar a proteção dupla. Se uma política AMS com o mesmo nome não existir, as mensagens serão colocadas no estado em que se encontram.

- Se houver uma política para a DLQ com a opção de tolerância no comando setmqspl configurada como desativada, isto é, '**-t** O', a colocação na DLQ falhará se a mensagem não for protegida pelo AMS e, portanto, não tiver um cabeçalho PDMQ. Isso acontece se a mensagem chegar ao receptor sem um cabeçalho PDMQ. Esse é o putter original da mensagem não tinha uma política para o destino, e o O receptor não possui SPLPROT (ASPOLICY) configurado.
- Um MCA poderá falhar ao enviar uma mensagem ao DLQ se a política do AMS definida para o DLQ não permitir o ID do usuário no qual o iniciador do canal está sendo executado para proteger a mensagem.
- Os canais receptores geralmente colocam mensagens não entregues na DLQ local, enquanto canais emissores geralmente colocam mensagens que não podem ser processadas por algum motivo, por exemplo, mensagens muito grandes para fila ou cabeçalho MQXQH inválido e assim por diante para a DLQ local.
- Os manipuladores DLQ geralmente só olham para o cabeçalho DLQ (DLH) e não a carga útil da mensagem em si. Assim, o fato de que a carga útil da mensagem pode ser protegida não impede os manipuladores de determinarem a razão pela qual a mensagem foi colocada na DLQ.
- Se uma DLQ não estiver definida, o canal:
  - se encerrará anormalmente (e irá para o estado de nova tentativa) se uma mensagem persistente não puder ser entregue;
  - descartará uma mensagem não entregue não persistente e continuará a execução.

Conceitos relacionados

"Manipulação de erros para o AMS" na página 616
O IBM MQ Advanced Message Security define uma fila de manipulação de erros para gerenciar mensagens que contêm erros ou mensagens que não podem ser desprotegidas.

Cenários do usuário para o AMS

Familiarize-se com cenários possíveis para entender quais metas de negócios é possível obter com o Advanced Message Security.

Windows *Guia de iniciação rápida para o AMS em plataformas Windows*

Use este guia para configurar rapidamente o Advanced Message Security (AMS) para fornecer a segurança de mensagens em plataformas Windows No momento em que você concluir, você terá criado um banco de dados de chaves para verificar identidades do usuário e definido políticas de assinatura/criptografia para o seu gerenciador de filas.

Antes de começar

É necessário ter pelo menos os recursos a seguir instalados em seu sistema:

- Servidor
- Kit de ferramentas de desenvolvimento (para os programas de amostra)
- Advanced Message Security (AMS)

Consulte os recursos do IBM MQ para os sistemas Windows para obter detalhes.

Para obter informações sobre como usar o comando **setmqenv** para inicializar o ambiente atual para que os comandos IBM MQ apropriados possam ser localizados e executados pelo sistema operacional, consulte setmqenv (configurar ambiente IBM MQ).

1. Criando um gerenciador de filas e uma fila

Sobre esta tarefa

Todos os exemplos a seguir usam uma fila nomeada TEST.Q para passar mensagens entre aplicativos. Advanced Message Security usa interceptores para assinar e criptografar mensagens no ponto em que elas entram na estrutura IBM MQ através da interface padrão do IBM MQ. A configuração básica é feita no IBM MQ e é configurada nas etapas a seguir.

É possível usar o IBM MQ Explorer para criar o gerenciador de filas QM_VERIFY_AMS e sua fila local chamada TEST.Q usando todas as configurações do assistente padrão ou é possível usar os comandos localizados em `C:\Program Files\IBM\MQ\bin` Lembre-se de que deve-se ser um membro do grupo de usuários `mqm` para executar os comandos administrativos a seguir.

Procedimento

1. Crie um gerenciador de filas

```
crtmqm QM_VERIFY_AMS
```

2. Iniciar o Gerenciador de Filas

```
strmqm QM_VERIFY_AMS
```

3. Crie uma fila chamada TEST.Q inserindo o comando a seguir em **runmqsc** para o gerenciador de filas QM_VERIFY_AMS

```
DEFINE QLOCAL(TEST.Q)
```

Resultados

Se o procedimento for concluído, o comando inserido em **runmqsc** irá exibir detalhes sobre TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Criando e autorizando usuários

Sobre esta tarefa

Há dois usuários que aparecem neste exemplo: `alice`, o emissor e bob, o receptor. Para usar a fila do aplicativo, esses usuários precisam receber autoridade para usá-la. Além disso, para usar com sucesso as políticas de proteção que vamos definir estes usuários devem ser concedidos acesso a algumas filas do sistema. Para obter mais informações sobre o comando **setmqaut**, consulte **setmqaut**.

Procedimento

1. Crie os dois usuários e assegure que HOMEPATH e HOMEDRIVE estejam configurados para ambos esses usuários.
2. Autorize os usuários para se conectarem ao gerenciador de filas e para trabalhar com a fila

```
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. É necessário também permitir que os dois usuários naveguem na fila de políticas do sistema e coloquem mensagens na fila de erros.

```
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
```

Atenção: O IBM MQ otimiza o desempenho armazenando em cache políticas para que você não tenha que procurar registros para obter detalhes da política no SYSTEM.PROTECTION.POLICY.QUEUE em todos os casos..

O IBM MQ não armazena em cache todas as políticas disponíveis. Se houver um grande número de políticas, o IBM MQ armazenará em cache um número limitado de políticas. Portanto, se o gerenciador de filas tiver um número baixo de políticas definidas, não haverá a necessidade de fornecer a opção de navegação para o SYSTEM.PROTECTION.POLICY.QUEUE.

No entanto, é necessário dar autoridade de navegação a essa fila nos casos em que há um grande número de políticas definidas ou quando você está usando clientes antigos. O SYSTEM.PROTECTION.ERROR.QUEUE é usado para colocar mensagens de erro geradas pelo código AMS. A autoridade put para essa fila é verificada apenas quando você tenta colocar uma mensagem de erro na fila. Sua autoridade put em relação à fila não é verificada quando você tenta colocar ou obter mensagens de uma fila protegida por AMS.

Resultados

Os usuários agora são criados e as autoridades necessárias concedidas a eles.

Como proceder a seguir

Para verificar se as etapas foram executadas corretamente, use as amostras `amqsput` e `amqsget` , conforme descrito na seção “7. Testando a configuração” na página 623

3. Criando banco de dados de chaves e certificados

Sobre esta tarefa

O interceptor requer a chave pública dos usuários de envio para criptografar a mensagem. Portanto, o banco de dados de chaves das identidades do usuário mapeado para chaves pública e privada deve ser criado. No sistema real, no qual os usuários e aplicativos são dispersos por meio de vários computadores, cada usuário teria seu próprio keystore privado. Da mesma forma, nesse guia, criamos banco de dados de chaves para `alice` e bob e compartilhamos os certificados de usuário entre eles.

Nota: Neste guia, usamos os aplicativos de amostra escritos em C conectando usando ligações locais. Se você planeja usar aplicativos Java usando ligações do cliente, deve-se criar um keystore e certificados JKS usando o Java **keytool** command V 9.4.0 V 9.4.0 ou o IBM MQ **runmqktool** command Para mais informações, consulte “Guia de iniciação rápida para o AMS com os clientes Java” na página 641. Para todas as outras linguagens e para os aplicativos Java usar ligações locais as etapas deste guia estão corretas.

Procedimento

1. Crie um novo banco de dados de chaves para o usuário `alice`..
   Por exemplo, emita o seguinte comando para criar o novo banco de dados de chaves:

```
runmqakm -keydb -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -type cms -pw
passw0rd -stash
```

   Nota:

   - Use uma senha forte para proteger o banco de dados..

- Inclua o parâmetro **-stash** para armazenar em arquivo stash a senha do banco de dados de chaves criptografada em um arquivo

2. Crie um novo certificado autoassinado para identificar o usuário `alice` para uso na criptografia..
   Por exemplo, emita o comando a seguir para criar um novo certificado autoassinado:

```
runmqakm -cert -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -stashed
-label Alice_Cert -dn "CN=alice, O=IBM, C=GB"
```

   Nota:

   - Para o propósito desse guia, estamos usando certificado autoassinado que pode ser criado sem o uso de uma Autoridade de Certificação. Para sistemas de produção, é aconselhável usar certificados assinados por uma Autoridade de Certificação.
   - O parâmetro **-label** especifica o nome para o certificado, que os interceptores procurarão para receber as informações necessárias
   - O parâmetro **-dn** especifica os detalhes do Nome Distinto (DN) para o certificado. O Nome Distinto deve ser exclusivo para cada usuário

3. Repita as etapas “1” na página 620 e “2” na página 621 para o usuário bob..

Resultados

Os dois usuários `alice` e bob agora possuem cada um certificado autoassinado.

4. Criando keystore.conf

Sobre esta tarefa

Deve-se apontar os interceptores do Advanced Message Security para o diretório no qual os bancos de dados de chave e certificados estão localizados. Isso é feito por meio do arquivo `keystore.conf`, que mantém essas informações em texto sem formatação. Cada usuário deve ter um arquivo `keystore.conf` separado na pasta `.mqs`. Esta etapa deve ser feita para ambos, `alice` e bob.

O conteúdo de `keystore.conf` deve ser dessa forma:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

Exemplo

Para este cenário, o conteúdo do `keystore.conf` será o seguinte:

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

Nota:

- O caminho para o arquivo keystore deve ser fornecido sem extensão de arquivo.
- O rótulo certificado pode incluir espaços, portanto "Alice_Cert" e "Alice_Cert " (com um espaço no final), por exemplo, são reconhecidos como rótulos de dois certificados diferentes. No entanto, para evitar confusão, é melhor não usar espaços no nome do rótulo.
- Existem os formatos de keystore a seguir: CMS (Cryptographic Message Sintaxe), JKS (Java Keystore) e JCEKS (Java Cryptographic Extension Keystore). Para obter informações adicionais, consulte “Estrutura do arquivo de configuração do keystore (keystore.conf) para AMS” na página 655.
- *%HOMEDRIVE%\%HOMEPATH%*\`.mqs\keystore.conf` (por exemplo, C:\Documents and Settings\alice\.mqs\keystore.conf) é o local padrão em que Advanced Message Security procura pelo arquivo `keystore.conf`. Para obter informações sobre como usar um local não padrão para o `keystore.conf`, consulte “Usando keystores e certificados com o AMS” na página 654.
- Para criar o diretório `.mqs`, deve-se usar o prompt de comando.

5. Compartilhando os certificados

Sobre esta tarefa

Compartilhe os certificados entre os dois bancos de dados de chaves para que cada usuário possa identificar com sucesso o outro. Isso é feito extraindo cada certificado público do usuário para um arquivo, que é, então, incluído no banco de dados de chaves do outro usuário.

Nota: Tome cuidado para usar a opção *extract* e não a opção *export*. *Extract* obtém a chave pública do usuário, enquanto *export* obtém a chave pública e a privada. Usar *export* por engano comprometeria completamente o seu aplicativo, transmitindo a sua chave privada.

Procedimento

1. Extraia o certificado identificando `alice` para um arquivo externo:

```
runmqakm -cert -extract -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -pw passw0rd
-label Alice_Cert -target alice_public.arm
```

2. Inclua o certificado para o keystore `bob's`:

```
runmqakm -cert -add -db "C:\Documents and Settings\bob\AMS\bobkey.kdb" -pw passw0rd -label
Alice_Cert -file alice_public.arm
```

3. Repita as etapas para bob:

```
runmqakm -cert -extract -db "C:\Documents and Settings\bob\AMS\bobkey.kdb" -pw passw0rd
-label Bob_Cert -target bob_public.arm
```

```
runmqakm -cert -add -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -pw passw0rd
-label Bob_Cert -file bob_public.arm
```

Resultados

Os dois usuários, `alice` e bob, agora podem identificar um ao outro com êxito como tendo criado e compartilhado certificados autoassinados.

Como proceder a seguir

Verifique se um certificado está no keystore procurando por ele usando a GUI ou executando os comandos a seguir que imprimem seus detalhes:

```
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. Definindo a política de filas

Sobre esta tarefa

Com o gerenciador de filas criado e os interceptores preparados para interceptar mensagens e acessar chaves de criptografia, podemos começar a definir políticas de proteção no `QM_VERIFY_AMS` usado o comando `setmqspl`. Consulte setmqspl para obter mais informações sobre este comando. Cada nome da política deve ser o mesmo que o nome da fila para a qual ela deve ser aplicada.

Exemplo

Este é um exemplo de uma política definida para a fila TEST.Q. No exemplo, as mensagens são assinadas com o algoritmo Deprecated SHA1 e criptografados com o algoritmo AES256. `alice` é o único emissor válido e bob é o único receptor das mensagens nesta fila:

```
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"
```

Nota: Os DNs correspondem exatamente aos especificados no respectivo certificado do usuário do banco de dados de chaves.

Como proceder a seguir

Para verificar a política que você definiu, emita o comando a seguir:

```
dspmqspl -m QM_VERIFY_AMS
```

Para imprimir os detalhes da política como um conjunto de comandos `setmqspl`, use a sinalização `-export`. Isso permite armazenar políticas já definidas:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Testando a configuração

Sobre esta tarefa

Ao executar programas diferentes em diferentes usuários é possível verificar se o aplicativo foi configurado corretamente.

Procedimento

1. Alterne o usuário para ser executado como `alice`

   Clique com o botão direito em `cmd.exe` e selecione **Executar como ...**. Quando solicitado, efetue login como o usuário `alice`.
2. Como o usuário `alice` coloque uma mensagem usando um aplicativo de amostra:

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. Digite o texto da mensagem e em seguida pressione Enter.
4. Alterne o usuário para ser executado como bob

   Abra outra janela clicando com o botão direito em `cmd.exe` e selecionando **Executar como...**. Quando solicitado, efetue login como o usuário bob.
5. Como o usuário bob obtenha uma mensagem usando um aplicativo de amostra:

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

Resultados

Se o aplicativo foi configurado corretamente para ambos os usuários, a mensagem do usuário `alice` será exibida quando o bob executar o aplicativo de obtenção.

8. Testando a criptografia

Sobre esta tarefa

Para verificar se a criptografia está ocorrendo conforme o esperado, crie uma fila de alias que faça referência à fila original TEST.Q. Esta fila de alias não terá uma política de segurança e, portanto, nenhum usuário terá as informações para decriptografar a mensagem e, portanto, os dados criptografados serão mostrados.

Procedimento

1. Usando o comando **runmqsc** no gerenciador de filas QM_VERIFY_AMS, crie uma fila de alias.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Conceda a bob acesso para procurar da fila de alias

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. Como o usuário alice, coloque outra mensagem usando um aplicativo de amostra como antes:

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. Como o usuário bob, procure a mensagem usando um aplicativo de amostra por meio da fila de alias dessa vez:

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. Como o usuário bob, obtenha a mensagem usando um aplicativo de amostra a partir da fila local:

```
amqsget TEST.Q QM_VERIFY_AMS
```

Resultados

A saída do aplicativo amqsbcg mostra os dados criptografados que estão na fila, provando que a mensagem foi criptografada.

Linux AIX *Guia de iniciação rápida para o AMS no AIX and Linux*

Use este guia para configurar rapidamente o Advanced Message Security para fornecer segurança de mensagens no AIX and Linux. No momento em que você concluir, você terá criado um banco de dados de chaves para verificar identidades do usuário e definido políticas de assinatura/criptografia para o seu gerenciador de filas.

Antes de começar

É necessário ter pelo menos os componentes a seguir instalados em seu sistema:

- Tempo de execução
- Servidor
- Programas de Amostra
- IBM Global Security Kit (GSKit)
- Advanced Message Security

Consulte os tópicos a seguir para os nomes dos componentes em cada plataforma específica:

- Linux IBM MQ componentes para Linux sistemas

- AIX IBM MQ componentes para AIX sistemas

1. Criando um gerenciador de filas e uma fila

Sobre esta tarefa

Todos os exemplos a seguir usam uma fila nomeada TEST.Q para passar mensagens entre aplicativos. Advanced Message Security usa interceptores para assinar e criptografar mensagens no ponto em que elas entram na estrutura IBM MQ através da interface padrão do IBM MQ. A configuração básica é feita no IBM MQ e é configurada nas etapas a seguir.

É possível usar o IBM MQ Explorer para criar o gerenciador de filas QM_VERIFY_AMS e sua fila local chamada TEST.Q usando todas as configurações padrão do assistente ou é possível usar os comandos localizados em *MQ_INSTALLATION_PATH*/bin. Lembre-se de que deve-se ser um membro do grupo de usuários mqm para executar os comandos administrativos a seguir.

Procedimento

1. Crie um gerenciador de filas

```
crtmqm QM_VERIFY_AMS
```

2. Iniciar o Gerenciador de Filas

```
strmqm QM_VERIFY_AMS
```

3. Crie uma fila chamada TEST.Q inserindo o comando a seguir em **runmqsc** para o gerenciador de filas QM_VERIFY_AMS

```
DEFINE QLOCAL(TEST.Q)
```

Resultados

Se o procedimento foi concluído com sucesso, o comando a seguir inserido em **runmqsc** irá exibir detalhes sobre TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Criando e autorizando usuários

Sobre esta tarefa

Há dois usuários que aparecem neste exemplo: alice, o emissor e bob, o receptor. Para usar a fila do aplicativo, esses usuários precisam receber autoridade para usá-la. Além disso, para usar com sucesso as políticas de proteção que vamos definir estes usuários devem ser concedidos acesso a algumas filas do sistema. Para obter mais informações sobre o comando **setmqaut**, consulte **setmqaut**.

Procedimento

1. Crie os dois usuários

```
useradd alice
```

```
useradd bob
```

2. Autorize os usuários para se conectarem ao gerenciador de filas e para trabalhar com a fila

```
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. É necessário também permitir que os dois usuários naveguem na fila de políticas do sistema e coloquem mensagens na fila de erros.

```
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
```

Atenção: O IBM MQ otimiza o desempenho armazenando em cache políticas para que você não tenha que procurar registros para obter detalhes da política no SYSTEM.PROTECTION.POLICY.QUEUE em todos os casos..

O IBM MQ não armazena em cache todas as políticas disponíveis. Se houver um grande número de políticas, o IBM MQ armazenará em cache um número limitado de políticas. Portanto, se o gerenciador de filas tiver um número baixo de políticas definidas, não haverá a necessidade de fornecer a opção de navegação para o SYSTEM.PROTECTION.POLICY.QUEUE.

No entanto, é necessário dar autoridade de navegação a essa fila nos casos em que há um grande número de políticas definidas ou quando você está usando clientes antigos. O SYSTEM.PROTECTION.ERROR.QUEUE é usado para colocar mensagens de erro geradas pelo código AMS. A autoridade put para essa fila é verificada apenas quando você tenta colocar uma mensagem de erro na fila. Sua autoridade put em relação à fila não é verificada quando você tenta colocar ou obter mensagens de uma fila protegida por AMS.

Resultados

Agora os grupos de usuários são criados e as autoridades requeridas concedidas a eles. Desse modo os usuários que estiverem designados a esses grupos também terão permissão para se conectar ao gerenciador de filas e colocar e obter da fila.

Como proceder a seguir

Para verificar se as etapas foram executadas corretamente, use as amostras `amqsput` e `amqsget` , conforme descrito na seção “8. Testando a criptografia” na página 630

3. Criando banco de dados de chaves e certificados

Sobre esta tarefa

Para criptografar a mensagem, o interceptor requer a chave privada do usuário de envio e a chave pública(s) do destinatário(s). Portanto, o banco de dados de chaves das identidades do usuário mapeado para chaves pública e privada deve ser criado. No sistema real, no qual os usuários e aplicativos são dispersos por meio de vários computadores, cada usuário teria seu próprio keystore privado. Da mesma forma, nesse guia, criamos banco de dados de chaves para `alice` e bob e compartilhamos os certificados de usuário entre eles.

Nota: Neste guia, usamos os aplicativos de amostra escritos em C conectando usando ligações locais. Se você planeja usar aplicativos Java usando as ligações de cliente, deve-se criar um keystore e certificados JKS usando o comando **`keytool`**, que é parte do JRE (consulte “Guia de iniciação rápida para o AMS com os clientes Java” na página 641 para obter mais detalhes). Para todas as outras linguagens e para os aplicativos Java usar ligações locais as etapas deste guia estão corretas.

Procedimento

1. Crie um novo banco de dados de chaves para o usuário `alice`

```
mkdir /home/alice/.mqs -p
```

```
runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
```

Nota:

- É aconselhável usar uma senha forte para proteger o banco de dados.
- O parâmetro **`stash`** armazena a senha no arquivo `key.sth`, que interceptores podem usar para abrir o banco de dados.

2. Assegure que o banco de dados chave é legível

```
chmod +r /home/alice/.mqs/alicekey.kdb
```

3. Crie um certificado que identifica o usuário `alice` para uso na criptografia

```
runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert -dn
"cn=alice,O=IBM,c=GB" -default_cert yes
```

Nota:

- Para o propósito desse guia, estamos usando certificado autoassinado que pode ser criado sem o uso de uma Autoridade de Certificação. Para sistemas de produção, é aconselhável não usar certificados autoassinados, mas em vez disso contar com certificados assinados por uma Autoridade de certificação.
- O parâmetro **`label`** especifica o nome para o certificado, que os interceptores consultarão para receber as informações necessárias.
- O parâmetro **DN** especifica os detalhes do **Nome Distinto** (DN), que deve ser exclusivo para cada usuário.

4. Agora criamos o banco de dados de chaves, é necessário configurar a propriedade dele e verificar se ele está ilegível para todos os outros usuários.

```
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. Repita as etapas 1-4 para o usuário bob

Resultados

Os dois usuários `alice` e bob agora possuem cada um certificado autoassinado.

4. Criando keystore.conf

Sobre esta tarefa

Deve-se apontar os interceptores do Advanced Message Security para o diretório no qual os bancos de dados de chave e certificados estão localizados. Isso é feito por meio do arquivo `keystore.conf`, que mantém essas informações em texto sem formatação. Cada usuário deve ter um arquivo `keystore.conf` separado na pasta `.mqs`. Esta etapa deve ser feita para ambos, `alice` e bob.

O conteúdo de `keystore.conf` deve ser dessa forma:

```
cms.keystore = dir/keystore_file
```

```
cms.certificate = certificate_label
```

Exemplo

Para este cenário, o conteúdo do `keystore.conf` será o seguinte:

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

Nota:

- O caminho para o arquivo keystore deve ser fornecido sem extensão de arquivo.
- Existem os formatos de keystore a seguir: CMS (Cryptographic Message Sintaxe), JKS (Java Keystore) e JCEKS (Java Cryptographic Extension Keystore). Para obter informações adicionais, consulte "Estrutura do arquivo de configuração do keystore (keystore.conf) para AMS" na página 655.
- `HOME/.mqs/keystore.conf` é o local padrão em que o Advanced Message Security procura o arquivo `keystore.conf`. Para obter informações sobre como usar um local não padrão para o `keystore.conf`, consulte "Usando keystores e certificados com o AMS" na página 654.

5. Compartilhando os certificados

Sobre esta tarefa

Compartilhe os certificados entre os dois bancos de dados de chaves para que cada usuário possa identificar com sucesso o outro. Isso é feito extraindo cada certificado público do usuário para um arquivo, que é, então, incluído no banco de dados de chaves do outro usuário.

Nota: Tome cuidado para usar a opção *extract* e não a opção *export*. *Extract* obtém a chave pública do usuário, enquanto *export* obtém a chave pública e a privada. Usar *export* por engano comprometeria completamente o seu aplicativo, transmitindo a sua chave privada.

Procedimento

1. Extraia o certificado identificando `alice` para um arquivo externo:

   ```
   runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
   -target alice_public.arm
   ```

2. Inclua o certificado para o keystore `bob's`:

   ```
   runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file
   alice_public.arm
   ```

3. Repita a etapa para bob:

   ```
   runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
   bob_public.arm
   ```

4. Inclua o certificado para bob para o keystore `alice's`:

   ```
   runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
   bob_public.arm
   ```

Resultados

Os dois usuários, `alice` e bob, agora podem identificar um ao outro com êxito como tendo criado e compartilhado certificados autoassinados.

Como proceder a seguir

Verifique se o certificado está no keystore executando os comandos a seguir que imprimem seus detalhes:

```
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. Definindo a política de filas

Sobre esta tarefa

Com o gerenciador de filas criado e os interceptores preparados para interceptar mensagens e acessar chaves de criptografia, podemos começar a definir políticas de proteção no `QM_VERIFY_AMS` usado o comando `setmqspl`. Consulte setmqspl para obter mais informações sobre este comando. Cada nome da política deve ser o mesmo que o nome da fila para a qual ela deve ser aplicada.

Exemplo

Este é um exemplo de uma política definida para a fila `TEST.Q`. Neste exemplo, as mensagens são assinadas pelo usuário `alice` usando o algoritmo 

SHA1 algoritmo e criptografados usando o algoritmo 256-bit AES. `alice` é o único emissor válido e bob é o único receptor das mensagens nesta fila:

```
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"
```

Nota: Os DNs correspondem exatamente aos especificados no respectivo certificado do usuário do banco de dados de chaves.

Como proceder a seguir

Para verificar a política que você definiu, emita o comando a seguir:

```
dspmqspl -m QM_VERIFY_AMS
```

Para imprimir os detalhes da política como um conjunto de comandos `setmqspl`, use a sinalização `-export`. Isso permite armazenar políticas já definidas:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Testando a configuração

Sobre esta tarefa

Ao executar programas diferentes em diferentes usuários é possível verificar se o aplicativo foi configurado corretamente.

Procedimento

1. Mude para o diretório que contém as amostras. Se o MQ estiver instalado em um local não padrão, este poderá estar em um local diferente.

```
cd /opt/mqm/samp/bin
```

2. Alterne o usuário para ser executado como `alice`

```
su alice
```

3. Como o usuário `alice`, coloque uma mensagem usando um aplicativo de amostra:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. Digite o texto da mensagem e em seguida pressione Enter.
5. Pare de executar como o usuário `alice`

```
exit
```

6. Alterne o usuário para ser executado como bob

```
su bob
```

7. Como o usuário bob, obtenha uma mensagem usando um aplicativo de amostra:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Resultados

Se o aplicativo foi configurado corretamente para ambos os usuários, a mensagem do usuário `alice` será exibida quando o bob executar o aplicativo de obtenção.

8. Testando a criptografia

Sobre esta tarefa

Para verificar se a criptografia está ocorrendo conforme o esperado, crie uma fila de alias que faça referência à fila original `TEST.Q`. Esta fila de alias não terá uma política de segurança e, portanto, nenhum usuário terá as informações para decriptografar a mensagem e, portanto, os dados criptografados serão mostrados.

Procedimento

1. Usando o comando **runmqsc** no gerenciador de filas QM_VERIFY_AMS, crie uma fila de alias.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Conceda a bob acesso para procurar da fila de alias

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. Como o usuário `alice`, coloque outra mensagem usando um aplicativo de amostra como antes:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. Como o usuário bob, procure a mensagem usando um aplicativo de amostra por meio da fila de alias dessa vez:

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. Como o usuário bob, obtenha a mensagem usando um aplicativo de amostra a partir da fila local:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Resultados

A saída do aplicativo amqsbcg mostrará os dados criptografados que estão na fila, provando que a mensagem foi criptografada.

z/OS *Example AMS configurations on z/OS*

This section provides example configurations of policies and certificates for Advanced Message Security queuing scenarios on z/OS.

See Configuring Advanced Message Security for z/OS for details on how you configure Advanced Message Security.

The examples cover the Advanced Message Security policies required, and the digital certificates that must exist relative to users and key rings. The examples assume that the users involved in the scenarios have been set up by following the instructions provided in Grant users resource permissions for Advanced Message Security.

See also server-to-server message channel interception examples.

z/OS *Local queuing of integrity-protected messages for AMS on z/OS*

This example details the Advanced Message Security policies and certificates needed to send and retrieve integrity-protected messages to and from a queue, local to the putting and getting applications.

The example queue manager and queue are:

```
BNK6        - Queue manager
FIN.XFER.Q7 - Local queue
```

These users are used:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

Create the user certificates

In this example, only one user certificate is needed. This is the sending user's certificate which is needed to sign integrity-protected messages. The sending user is 'TELLER5'.

The Certificate Authority (CA) certificate is also required. The CA certificate is the certificate of the authority that issued the user's certificate. This can be a chain of certificates. If so, all certificates in the chain are required in the key ring of the Advanced Message Security task user, in this case user WMQBNK6.

A CA certificate can be created using the RACF RACDCERT command. This certificate is used to issue user certificates. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

This RACDCERT command creates a CA certificate which can then be used to issue a user certificate for user 'TELLER5'. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Your installation will have procedures for choosing or creating a CA certificate, as well as procedures for issuing certificates and distributing them to relevant systems.

When exporting and importing these certificates, Advanced Message Security requires:

- The CA certificate (chain).
- The user certificate and its private key.

If you are using RACF, the RACDCERT EXPORT command can be used to export certificates to a data set, and the RACDCERT ADD command can be used to import certificates from the data set. For more information about these and other RACDCERT commands, refer to *z/OS: Security Server RACF Command Language Reference*.

The certificates in this case, are required on the z/OS system running queue manager BNK6.

When the certificates have been imported on the z/OS system running BNK6, the user certificate requires the TRUST attribute. The RACDCERT ALTER command can be used to add the TRUST attribute to the certificate. For example:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

In this example, no certificate is required for the recipient user.

Connect certificates to relevant key rings

When the required certificates have been created or imported, and set as trusted, they must be connected to the appropriate user key rings on the z/OS system running BNK6. To create the key rings use the RACDCERT ADDRING commands:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

This creates a key ring for the Advanced Message Security task user, WMQBNK6, and a key ring for the sending user, 'TELLER5'. Note that the key ring name drq.ams.keyring is mandatory, and the name is case-sensitive.

When the key rings have been created, the relevant certificates can be connected:

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

The sending user certificate must be connected as DEFAULT. If the sending user has more than one certificate in its drq.ams.keyring, the default certificate is used for signing purposes.

The creation and modification of certificates is not recognized by Advanced Message Security until the queue manager is stopped and restarted, or the z/OS **MODIFY** command is used to refresh the Advanced Message Security certificate configuration. For example:

```
F BNK6AMSM,REFRESH KEYRING
```

Create the Advanced Message Security policy

In this example, integrity-protected messages are put to queue FIN.XFER.Q7 by an application running as user 'TELLER5', and retrieved from the same queue by an application running as user 'FINADM2', so only one Advanced Message Security policy is required.

Advanced Message Security policies are created using the CSQ0UTIL utility that is documented at The message security policy utility (CSQ0UTIL).

Use the CSQ0UTIL utility to run the following command:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK6. The policy name and associated queue is FIN.XFER.Q7. The algorithm that is used to generate the sender's signature is MD5, and the distinguished name (DN) of the sending user is 'CN=Teller5,O=BCO,C=US'.

After defining the policy, either restart the BNK6 queue manager, or use the z/OS **MODIFY** command to refresh the Advanced Message Security policy configuration. For example:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *Enfileiramento local de mensagens protegidas por privacidade para o AMS no z/OS*

Este exemplo detalha as políticas e os certificados do Advanced Message Security necessários para enviar e recuperar mensagens protegidas por privacidade para/de uma fila local para os aplicativos de colocação e obtenção. As mensagens de privacidade protegida são assinadas e criptografadas.

O exemplo do gerenciador de filas e a fila local são os seguintes:

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

Esses usuários são usados:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

As etapas para configurar esse cenário são:

Criar os certificados de usuário

Nesse exemplo, dois certificados de usuário são necessários. Esses são certificados de usuário de envio que é necessário para assinar mensagens e o certificado de usuário destinatário, que é necessário para criptografar e decriptografar os dados da mensagem. O usuário de envio é 'TELLER5' e o usuário destinatário é 'FINADM2'.

O certificado da autoridade de certificação (CA) também é necessário.O certificado de autoridade de certificação é o certificado da autoridade que emitiu o certificado de usuário. Isso pode ser uma cadeia de certificados. Se assim for, todos os certificados na cadeia são necessários no conjunto de chaves do usuário da tarefa do Advanced Message Security, neste caso o usuário WMQBNK6.

Um certificado de autoridade de certificação pode ser criado usando o comando RACDCERT do RACF. Este certificado é usado para emitir certificados de usuário. Por exemplo:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Este comando RACDCERT cria um certificado de autoridade de certificação que pode então ser usado para emitir certificados de usuário para os usuários TELLER5' e 'FINADM2'. Por exemplo:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

```
RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

A instalação terá procedimentos para escolher ou criar um certificado de autoridade de certificação, bem como os procedimentos para emitir certificados e distribuí-los para sistemas relevantes.

Ao exportar e importar esses certificados, o Advanced Message Security requer:

- O certificado de autoridade de certificação (cadeia).
- O certificado de usuário de envio e sua chave privada.
- O certificado de usuário destinatário e sua chave privada.

Se você estiver usando o RACF, o comando RACDCERT EXPORT pode ser usado para exportar certificados para um conjunto de dados e o comando RACDCERT ADD pode ser usado para importar certificados do conjunto de dados. Para obter mais informações sobre esses e outros comandos RACDCERT, consulte RACDCERT (Gerenciar RACF certificados digitais) no *z/OS: Security Server RACF Command Language Reference*.

Os certificados nesse caso são necessários no sistema z/OS que está executando o gerenciador de filas BNK6.

Quando os certificados forem importados no sistema z/OS que estiver executando BNK6, os certificados de usuário irão requerer o atributo TRUST. O comando RACDCERT ALTER pode ser usado para incluir o atributo TRUST ao certificado. Por exemplo:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Conectar os certificados aos conjuntos de chaves relevantes

Quando os certificados necessários forem criados ou importados e definidos como confiáveis, eles devem ser conectados aos conjuntos de chaves apropriados do usuário no sistema z/OS que estiver executando o BNK6. Para criar os conjuntos de chaves use o comando RACDCERT ADDRING:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Isso cria um conjunto de chaves para o usuário da tarefa do Advanced Message Security e conjuntos de chaves para os usuários de envio e destinatário. Observe que o nome do conjunto de chaves `drq.ams.keyring` é obrigatório e o nome faz distinção entre maiúsculas e minúsculas.

Quando os conjuntos de chave forem criados, os certificados relevantes poderão ser conectados.

```
RACDCERT ID(WMQBNK6) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

```
RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))
```

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

```
RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

Os certificados de usuário de envio e destinatário devem estar conectados como DEFAULT. Se qualquer usuário possuir mais de um certificado em seu drq.ams.keyring, o certificado padrão será usado para propósitos de assinatura e decriptografia.

O certificado de usuário destinatário deve também estar conectado ao conjunto de chaves do usuário de tarefa do Advanced Message Security com USAGE(SITE). Isso ocorre porque a tarefa Advanced Message Security precisa da chave pública do destinatário ao criptografar os dados da mensagem. O USAGE(SITE) impede a chave privada de ser acessível no conjunto de chaves.

A criação e modificação de certificados não é reconhecida pelo Advanced Message Security até que o gerenciador de filas seja parado e reiniciado ou o comando do z/OS **MODIFY** seja usado para atualizar a configuração de certificado do Advanced Message Security. Por exemplo:

```
F BNK6AMSM,REFRESH KEYRING
```

Criar a política do Advanced Message Security

Neste exemplo, as mensagens de privacidade protegida são colocadas na fila FIN.XFER.Q8 por um aplicativo em execução como o usuário 'TELLER5' e recuperadas a partir da mesma fila por um aplicativo em execução como o usuário 'FINADM2', portanto, apenas uma política do Advanced Message Security é necessária.

As políticas do Advanced Message Security são criadas usando o utilitário CSQ0UTIL, que está documentado em O utilitário de política de segurança da mensagem (CSQ0UTIL).

Use o utilitário CSQ0UTIL para executar o comando a seguir:

```
setmqspl -m BNK6 -p FIN.XFER.Q8 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

Nesta política, o gerenciador de filas é identificado como BNK6. O nome da política e a fila associada é FIN.XFER.Q8. O algoritmo usado para gerar a assinatura do emissor é Deprecated SHA1e o nome distinto (DN) do usuário de envio é 'CN=Teller5,O=BCO,C=US' e o usuário destinatário é 'CN=FinAdm2,O=BCO,C=US'. O algoritmo usado para criptografar os dados da mensagem é Deprecated 3DES.

Depois de definir a política, reinicie o gerenciador de filas BNK6 ou use o comando do z/OS **MODIFY** para atualizar a configuração de política do Advanced Message Security. Por exemplo:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *Remote queuing of integrity-protected messages for AMS on z/OS*

This example details the Advanced Message Security policies and certificates needed to send and retrieve integrity-protected messages to and from queues managed by two different queue managers. The two queue managers can be running on the same z/OS system, or on different z/OS systems, or one queue manager can be on a distributed system running Advanced Message Security.

The example queue managers and queues are:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Note: For this example, BNK6 and BNK7 are queue managers running on different z/OS systems.

These users are used:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMStask user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

The steps to configure this scenario are as follows:

Create the user certificates

In this example, only one user certificate is needed. This is the sending user's certificate which is needed to sign integrity-protected message. The sending user is 'TELLER5'.

The Certificate Authority (CA) certificate is also required. The CA certificate is the certificate of the authority that issued the user's certificate. This can be a chain of certificates. If so, all certificates in the chain are required in the key ring of the Advanced Message Security task user, in this case user WMQBNK7.

A CA certificate can be created using the RACF RACDCERT command. This certificate is used to issue user certificates. For example:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

This RACDCERT command creates a CA certificate which can then be used to issue user certificate for user 'TELLER5'. For example:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Your installation will have procedures for choosing or creating a CA certificate, as well as procedures for issuing certificates and distributing them to relevant systems.

When exporting and importing these certificates, Advanced Message Security require:

- The CA certificate (chain).
- The sending user certificate and its private key.

If you are using RACF, the RACDCERT EXPORT command can be used to export certificates to a data set, and the RACDCERT ADD command can be used to import certificates from the data set. For more information about these and other RACDCERT commands, refer to RACDCERT (Manage RACF digital certificates) in the *z/OS: Security Server RACF Command Language Reference*.

The certificates in this case, are required on the z/OS system running queue manager BNK6 and BNK7.

In this example, the sending certificate must be imported on the z/OS system running BNK6, and the CA certificate must be imported on the z/OS system running BNK7. When the certificates have been imported, the user certificate requires the TRUST attribute. The RACDCERT ALTER command can be used to add the TRUST attribute to the certificate. For example, on BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

Connect certificates to relevant key rings

When the required certificates have been created or imported, and set as trusted, they must be connected to the appropriate user key rings on the z/OS system running BNK6 and BNK7.

To create the key rings use the RACDCERT ADDRING command, on BNK6:

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

This creates a key ring for the sending user on BNK6. Note that the key ring name drq.ams.keyring is mandatory, and the name is case-sensitive.

On BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

This creates a key ring for the Advanced Message Security task user on BNK7. No user key ring is required for 'TELLER5' on BNK7.

When the key rings have been created, the relevant certificates can be connected.

On BNK6:

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

On BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

The sending user certificate must be connected as DEFAULT. If the sending user has more than one certificate in its drq.ams.keyring, the default certificate is used for signing purposes.

The creation and modification of certificates is not recognized by Advanced Message Security until the queue manager is stopped and restarted, or the z/OS **MODIFY** command is used to refresh the Advanced Message Security certificate configuration. For example:

On BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

On BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Create the Advanced Message Security policies

In this example, integrity-protected messages are put to remote queue FIN.XFER.Q7 on BNK6 by an application running as user 'TELLER5', and retrieved from local queue FIN.RCPT.Q7 on BNK7 by an application running as user 'FINADM2', so two Advanced Message Security policies are required.

Advanced Message Security policies are created using the CSQ0UTIL utility that is documented at The message security policy utility (CSQ0UTIL).

Use the CSQ0UTIL utility to run the following command to define an integrity policy for the remote queue on BNK6:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK6. The policy name and associated queue is FIN.XFER.Q7. The algorithm that is used to generate the sender's signature is MD5, and the distinguished name (DN) of the sending user is 'CN=Teller5,O=BCO,C=US'.

Also, use the CSQ0UTIL utility to run the following command to define an integrity policy for the local queue on BNK7:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

In this policy, the queue manager is identified as BNK7. The policy name and associated queue is FIN.RCPT.Q7. The algorithm expected for the sender's signature is MD5, and the distinguished name (DN) of the sending user is expected to be 'CN=Teller5,O=BCO,C=US'.

After defining the two policies, either restart the BNK6 and BNK7 queue managers, or use the z/OS **MODIFY** command to refresh the Advanced Message Security policy configurations. For example:

On BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

On BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

z/OS *Enfileiramento remoto de mensagens protegidas por privacidade para o AMS no z/OS*

Este exemplo detalha as políticas e certificados do Advanced Message Security necessários para enviar e recuperar mensagens de privacidade protegida para e a partir de filas gerenciadas por dois gerenciadores de filas diferentes. Os dois gerenciadores de filas podem estar em execução no mesmo sistema z/OS

ou em diferentes sistemas z/OS ou um gerenciador de filas pode estar em um sistema distribuído executando o Advanced Message Security.

Os gerenciadores de filas e filas de exemplo são:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Nota: Para este exemplo, BNK6 e BNK7 são gerenciadores de filas em execução em diferentes sistemas z/OS com o mesmo nome.

Esses usuários são usados:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

As etapas para configurar esse cenário são as seguintes:

Criar os certificados de usuário

Nesse exemplo, dois certificados de usuário são necessários. Esses são certificados de usuário de envio que é necessário para assinar mensagens e o certificado de usuário destinatário, que é necessário para criptografar e decriptografar os dados da mensagem. O usuário de envio é 'TELLER5' e o usuário destinatário é 'FINADM2'.

O certificado da autoridade de certificação (CA) também é necessário.O certificado de autoridade de certificação é o certificado da autoridade que emitiu o certificado de usuário. Isso pode ser uma cadeia de certificados. Se assim for todos os certificados na cadeia são necessários no conjunto de chaves do usuário da tarefa do Advanced Message Security, neste caso o usuário WMQBNK7.

Um certificado de autoridade de certificação pode ser criado usando o comando RACDCERT do RACF. Este certificado é usado para emitir certificados de usuário. Por exemplo:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Este comando RACDCERT cria um certificado de autoridade de certificação que pode então ser usado para emitir certificados de usuário para os usuários TELLER5' e 'FINADM2'. Por exemplo:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

```
RACDCERT ID(FINADM2) GENCERT SUBJECTSDN(CN('FinAdm2') O('BCO') C('US'))
WITHLABEL('FinAdm2') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

A instalação terá procedimentos para escolher ou criar um certificado de autoridade de certificação, bem como os procedimentos para emitir certificados e distribuí-los para sistemas relevantes.

Ao exportar e importar esses certificados, o Advanced Message Security requer:

- O certificado de autoridade de certificação (cadeia).
- O certificado de usuário de envio e sua chave privada.
- O certificado de usuário destinatário e sua chave privada.

Se você estiver usando o RACF, o comando RACDCERT EXPORT pode ser usado para exportar certificados para um conjunto de dados e o comando RACDCERT ADD pode ser usado para importar certificados do conjunto de dados.

Para obter mais informações sobre esses e outros comandos RACDCERT, consulte RACDCERT (Gerenciar RACF certificados digitais) no *z/OS: Security Server RACF Command Language Reference*.

Os certificados, neste caso, são necessários no sistema z/OS que está executando o gerenciador de filas BNK6 e BNK7.

Neste exemplo, os certificados de envio e destinatário devem ser importados no sistema z/OS que está executando o BNK6 e os certificados de CA e destinatário devem ser importados no sistema z/OS que está executando o BNK7. Quando os certificados forem importados, os certificados de usuário irão requerer o atributo TRUST. O comando RACDCERT ALTER pode ser usado para incluir o atributo TRUST ao certificado. Por exemplo:

No BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

No BNK7:

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Conectar os certificados aos conjuntos de chaves relevantes

Quando os certificados necessários forem criados ou importados e definidos como confiáveis, eles devem ser conectados aos conjuntos de chaves apropriados do usuário nos sistemas z/OS que estão executando BNK6 e BNK7.

Para criar os conjuntos de chaves use o comando RACDCERT ADDRING:

No BNK6:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Isso cria um conjunto de chaves para o usuário da tarefa do Advanced Message Security e um conjunto de chaves para o usuário de envio no BNK6. Observe que o nome do conjunto de chaves drq.ams.keyring é obrigatório e o nome faz distinção entre maiúsculas e minúsculas.

No BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Isso cria um conjunto de chaves para o usuário da tarefa do Advanced Message Security e um conjunto de chaves para o destinatário do usuário no BNK7.

Quando os conjuntos de chave forem criados, os certificados relevantes poderão ser conectados.

No BNK6:

```
RACDCERT ID(WMQBNK6) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) USAGE(SITE))
```

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

No BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

```
RACDCERT ID(FINADM2) CONNECT(ID(FINADM2) LABEL('FinAdm2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

Os certificados de usuário de envio e destinatário devem estar conectados como DEFAULT. Se qualquer usuário possui mais de um certificado em seu drq.ams.keyring, o certificado padrão é usado para propósitos de assinatura e criptografia/decriptografia.

No BNK6, o certificado de usuário destinatário também deve estar conectado ao conjunto de chaves do usuário da tarefa do Advanced Message Security com USAGE(SITE). Isso ocorre porque a tarefa Advanced Message Security precisa da chave pública do destinatário ao criptografar os dados da mensagem. O USAGE(SITE) impede a chave privada de ser acessível no conjunto de chaves.

A criação e modificação de certificados não é reconhecida pelo Advanced Message Security até que o gerenciador de filas seja parado e reiniciado ou o comando do z/OS **MODIFY** seja usado para atualizar a configuração de certificado do Advanced Message Security. Por exemplo:

No BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

No BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Criar as políticas do Advanced Message Security

Neste exemplo, as mensagens de privacidade protegida são colocadas na fila remota FIN.XFER.Q7 no BNK6 por um aplicativo em execução como usuário o 'TELLER5' e recuperadas da fila local FIN.RCPT.Q7 no BNK7 por um aplicativo em execução como o usuário 'FINADM2', portanto, duas políticas do Advanced Message Security são necessárias.

As políticas do Advanced Message Security são criadas usando o utilitário CSQ0UTIL, que está documentado em O utilitário de política de segurança da mensagem (CSQ0UTIL).

Use o utilitário CSQ0UTIL para executar o comando a seguir para definir uma política de privacidade para a fila remota no BNK6:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

Nesta política, o gerenciador de filas é identificado como BNK6. O nome da política e da fila associada é FIN.XFER.Q7. O algoritmo usado para gerar a assinatura do emissor é Deprecated SHA1, o nome distinto (DN) do usuário de envio é 'CN=Teller5,O=BCO,C=US' e o usuário destinatário é 'CN=FinAdm2,O=BCO,C=US'. O algoritmo usado para criptografar os dados da mensagem é Deprecated 3DES.

Além disso, use o utilitário CSQ0UTIL para executar o comando a seguir para definir uma política de privacidade para a fila local no BNK7:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s SHA1 -e 3DES -a CN=Teller5,O=BCO,C=US -r
CN=FinAdm2,O=BCO,C=US
```

Nesta política, o gerenciador de filas é identificado como BNK7. O nome da política e da fila associada é FIN.RCPT.Q7. O algoritmo esperado para a assinatura do emissor é Deprecated SHA1, o nome distinto (DN) do usuário de envio deve ser 'CN=Teller5,O=BCO,C=US' e o usuário destinatário é 'CN=FinAdm2,O=BCO,C=US'. O algoritmo usado para decriptografar os dados da mensagem é Deprecated 3DES.

Depois de definir as duas políticas, reinicie os gerenciadores de filas BNK6 e BNK7 ou use o comando do z/OS **MODIFY** para atualizar a configuração de política do Advanced Message Security. Por exemplo:

No BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

No BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

Guia de iniciação rápida para o AMS com os clientes Java

Use este guia para configurar rapidamente o Advanced Message Security para fornecer segurança de mensagens para aplicativos do Java conectando usando ligações do cliente. No momento em que você concluir, você terá criado um armazenamento de chaves para verificar identidades do usuário e definido políticas de assinatura/criptografia para o seu gerenciador de filas.

Antes de começar

Assegure-se de ter os componentes apropriados instalados, conforme descrito em "Guia de iniciação rápida para o AMS em plataformas Windows" na página 618 ou "Guia de iniciação rápida para o AMS no AIX and Linux" na página 624.

1. Criando um gerenciador de filas e uma fila

Sobre esta tarefa

Todos os exemplos a seguir usam uma fila nomeada TEST.Q para passar mensagens entre aplicativos. Advanced Message Security usa interceptores para assinar e criptografar mensagens no ponto em que elas entram na estrutura IBM MQ através da interface padrão do IBM MQ. A configuração básica é feita no IBM MQ e é configurada nas etapas a seguir.

Procedimento

1. Crie um gerenciador de filas

   ```
   crtmqm QM_VERIFY_AMS
   ```

2. Iniciar o Gerenciador de Filas

   ```
   strmqm QM_VERIFY_AMS
   ```

3. Crie e inicie um listener inserindo os comandos a seguir no **runmqsc** para o gerenciador de filas QM_VERIFY_AMS

   ```
   DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
   ```

   ```
   START LISTENER(AMS.LSTR)
   ```

4. Crie um canal para os nossos aplicativos para se conectar inserindo o comando a seguir em **runmqsc** para o gerenciador de filas QM_VERIFY_AMS

   ```
   DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
   ```

5. Crie uma fila chamada TEST.Q inserindo o comando a seguir em **runmqsc** para o gerenciador de filas QM_VERIFY_AMS

   ```
   DEFINE QLOCAL(TEST.Q)
   ```

Resultados

Se o procedimento concluiu com sucesso, o comando a seguir inserido em **runmqsc** exibe detalhes sobre TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Criando e autorizando usuários

Sobre esta tarefa

Há dois usuários que aparecem neste cenário: `alice`, o emissor e bob, o receptor. Para usar a fila do aplicativo, esses usuários precisam receber autoridade para usá-la. Além disso, para usar com sucesso as políticas de proteção definidas neste cenário, esses usuários devem ser concedidos acesso a algumas filas do sistema. Para obter mais informações sobre o comando **setmqaut**, consulte **setmqaut**.

Procedimento

1. Crie os dois usuários conforme descrito no **Guia de iniciação rápida** (Windows ou AIX and Linux) para sua plataforma.
2. Autorize os usuários para se conectarem ao gerenciador de filas e para trabalhar com a fila

   ```
   setmqaut -m QM_VERIFY_AMS -t qmgr -p alice -p bob +connect +inq
   ```

   ```
   setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p alice +put
   ```

   ```
   setmqaut -m QM_VERIFY_AMS -n TEST.Q -t queue -p bob +get +inq +browse
   ```
3. É necessário também permitir que os dois usuários naveguem na fila de políticas do sistema e coloquem mensagens na fila de erros.

   ```
   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
   ```

   Atenção: O IBM MQ otimiza o desempenho armazenando em cache políticas para que você não tenha que procurar registros para obter detalhes da política no SYSTEM.PROTECTION.POLICY.QUEUE em todos os casos..

   O IBM MQ não armazena em cache todas as políticas disponíveis. Se houver um grande número de políticas, o IBM MQ armazenará em cache um número limitado de políticas. Portanto, se o gerenciador de filas tiver um número baixo de políticas definidas, não haverá a necessidade de fornecer a opção de navegação para o SYSTEM.PROTECTION.POLICY.QUEUE.

   No entanto, é necessário dar autoridade de navegação a essa fila nos casos em que há um grande número de políticas definidas ou quando você está usando clientes antigos. O SYSTEM.PROTECTION.ERROR.QUEUE é usado para colocar mensagens de erro geradas pelo código AMS. A autoridade put para essa fila é verificada apenas quando você tenta colocar uma mensagem de erro na fila. Sua autoridade put em relação à fila não é verificada quando você tenta colocar ou obter mensagens de uma fila protegida por AMS.

Resultados

Os usuários agora são criados e as autoridades necessárias concedidas a eles.

Como proceder a seguir

Para verificar se as etapas foram realizadas corretamente, use as amostras `JmsProducer` e `JmsConsumer` conforme descrito na seção "7. Testando a configuração" na página 646.

3. Criando banco de dados de chaves e certificados

Sobre esta tarefa

Para criptografar a mensagem para o interceptor requer a chave pública dos usuários de envio. Portanto, o banco de dados de chaves das identidades do usuário mapeado para chaves pública e privada deve ser criado. No sistema real, no qual os usuários e aplicativos são dispersos por meio de vários computadores, cada usuário teria seu próprio keystore. Da mesma forma, nesse guia, criamos banco de dados de chaves para `alice` e bob e compartilhamos os certificados de usuário entre eles.

Nota: Neste guia, podemos usar os aplicativos de amostra escritos em Java conectando usando as ligações de cliente. Se você planeja usar os aplicativos Java usando ligações locais ou aplicativos C, deve-se criar um keystore CMS e certificados usando o comando **runmqakm**. Para obter mais informações, consulte "Guia de iniciação rápida para o AMS em plataformas Windows" na página 618 e "Guia de iniciação rápida para o AMS no AIX and Linux" na página 624.

Procedimento

1. Crie um diretório no qual criar seu keystore, por exemplo `/home/alice/.mqs`. Talvez você queira criá-lo no mesmo diretório usado pelo Guia de Iniciação Rápida para sua plataforma.. Para obter mais informações, consulte o "Guia de iniciação rápida para o AMS em plataformas Windows" na página 618 e o "Guia de iniciação rápida para o AMS no AIX and Linux" na página 624.

   Nota: Esse diretório é referido como *keystore-dir* nas etapas a seguir
2. Crie um novo keystore e o certificado que identifica o usuário `alice` para uso na criptografia

   Nota: O comando **keytool** é parte do JRE.

   ```
   keytool -genkey -alias Alice_Java_Cert -keyalg RSA -keystore keystore-dir/keystore.jks
   -storepass passw0rd
   -dname "CN=alice, O=IBM, C=GB" -keypass passw0rd
   ```

   Nota:

   - Se o *keystore-dir* contiver espaços, deve-se colocar entre aspas ao redor o nome completo de seu keystore
   - É aconselhável usar uma senha forte para proteger o keystore.
   - Para o propósito desse guia, estamos usando certificado autoassinado que pode ser criado sem o uso de uma Autoridade de Certificação. Para sistemas de produção, é aconselhável não usar certificados autoassinados, mas em vez disso contar com certificados assinados por uma Autoridade de certificação.
   - O parâmetro **alias** especifica o nome para o certificado, que os interceptores consultarão para receber as informações necessárias.
   - O parâmetro **dname** especifica os detalhes do **Nome Distinto** (DN), que deve ser exclusivos para cada usuário.
3. No AIX and Linux, assegure-se de que o keystore esteja legível

   ```
   chmod +r keystore-dir/keystore.jks
   ```
4. Repita as etapas 1-4 para o usuário bob

Resultados

Os dois usuários `alice` e bob agora possuem cada um certificado autoassinado.

4. Criando keystore.conf

Sobre esta tarefa

Deve-se apontar os interceptores do Advanced Message Security para o diretório no qual os bancos de dados de chave e certificados estão localizados. Isso é feito por meio do arquivo `keystore.conf`, que retém essas informações em texto sem formatação. Cada usuário deve ter um arquivo `keystore.conf` separado. Esta etapa deve ser feita para ambos `alice` e bob.

Exemplo

Para este cenário, os conteúdos do `keystore.conf` para `alice` são os seguintes:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Alice_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

Para este cenário, os conteúdos do `keystore.conf` para bob são os seguintes:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Bob_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

Nota:

- O caminho para o arquivo keystore deve ser fornecido sem extensão de arquivo.
- Se você já tiver um arquivo `keystore.conf` porque seguiu as instruções no Guia de iniciação rápida (Windows ou AIX and Linux), poderá editar o arquivo existente para incluir essas linhas.
- Para obter informações adicionais, consulte "Estrutura do arquivo de configuração do keystore (keystore.conf) para AMS" na página 655.

5. Compartilhando os certificados

Sobre esta tarefa

Compartilhe os certificados entre os dois keystores para que cada usuário pode identificar com sucesso o outro. Isso é feito extraindo cada certificado do usuário e importando-o para o keystore do outro usuário.

Importante: Os termos *extract* e *export* são usados de forma diferente por diferentes comandos de gerenciamento de certificado

- O comando IBM Global Security Kit (GSKit) **runmqakm** usa o termo *extract* para se referir ao processo de copiar apenas a parte pública de um certificado de um keystore e o termo *export* para se referir ao processo de copiar certificados e suas chaves públicas e privadas associadas de um keystore para outro.
- O comando Java **keytool** V 9.4.0 V 9.4.0 e o comando IBM MQ **runmqktool** usam o termo *export* para se referir ao processo de copiar apenas a parte pública de um certificado de um keystore.

Essa distinção é importante, pois usar *export* incorretamente pode comprometer seu aplicativo expondo sua chave privada. Como a distinção é tão importante, a documentação do IBM MQ usa esses termos consistentemente. Por esses motivos, o procedimento a seguir refere-se a *extrair* certificados usando a opção `exportcert` no comando **keytool** .

Procedimento

1. Extraia o certificado que identifica `alice`.

```
keytool -exportcert -keystore alice-keystore-dir/keystore.jks -storepass passw0rd
-alias Alice_Java_Cert -file alice-keystore-dir/Alice_Java_Cert.cer
```

2. Importe o certificado identificando `alice` no keystore que bob usará. Quando solicitado indique que você irá confiar nesse certificado.

```
keytool -importcert -file alice-keystore-dir/Alice_Java_Cert.cer -alias Alice_Java_Cert
-keystore bob-keystore-dir/keystore.jks -storepass passw0rd
```

3. Repita as etapas para bob

Resultados

Os dois usuários, `alice` e bob, agora podem identificar um ao outro com êxito como tendo criado e compartilhado certificados autoassinados.

Como proceder a seguir

Verifique se o certificado está no keystore executando os comandos a seguir que imprimem seus detalhes:

```
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. Definindo a política de filas

Sobre esta tarefa

Com o gerenciador de filas criado e os interceptores preparados para interceptar mensagens e acessar chaves de criptografia, podemos começar a definir políticas de proteção no `QM_VERIFY_AMS` usado o comando `setmqspl`. Consulte setmqspl para obter mais informações sobre este comando. Cada nome da política deve ser o mesmo que o nome da fila para a qual ela deve ser aplicada.

Exemplo

Este é um exemplo de uma política definida na fila `TEST.Q`, assinada pelo usuário `alice` usando o algoritmo 

SHA1 e criptografada usando o algoritmo 256-bit AES para o usuário bob:

```
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"
```

Nota: Os DNs correspondem exatamente aos especificados no respectivo certificado do usuário do banco de dados de chaves.

Como proceder a seguir

Para verificar a política que você definiu, emita o comando a seguir:

```
dspmqspl -m QM_VERIFY_AMS
```

Para imprimir os detalhes da política como um conjunto de comandos `setmqspl`, o sinalizador `-export`. Isso permite armazenar políticas já definidas:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

Antes de começar

Assegure-se de que a versão do Java que você está usando tenha os arquivos de políticas JCE irrestritos instalados.

Nota: A versão do Java fornecida na instalação do IBM MQ já possui esses arquivos de políticas Ela pode ser localizada em *MQ_INSTALLATION_PATH*/java/bin.

Sobre esta tarefa

Ao executar programas diferentes em diferentes usuários é possível verificar se o aplicativo foi configurado corretamente. Para obter mais informações sobre como executar programas em usuários diferentes, consulte "Guia de iniciação rápida para o AMS em plataformas Windows" na página 618 e "Guia de iniciação rápida para o AMS no AIX and Linux" na página 624.

Procedimento

1. Para executar estes aplicativos de amostra do JMS, use a configuração de CLASSPATH para sua plataforma conforme mostrado em Variáveis de ambiente usadas pelo IBM MQ classes for JMS para assegurar que o diretório de amostras está incluído.
2. Como o usuário `alice`, coloque uma mensagem usando um aplicativo de amostra, se conectando como um cliente:

   ```
   java JmsProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

3. Como o usuário bob, obtenha uma mensagem usando um aplicativo de amostra, se conectando como um cliente:

   ```
   java JmsConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

Resultados

Se o aplicativo foi configurado corretamente para ambos os usuários, a mensagem do usuário `alice` será exibida quando o bob executar o aplicativo de obtenção.

Protegendo filas remotas no AMS

Para proteger completamente as filas remotas, as políticas devem ser configuradas na fila remota e na fila local para a qual as mensagens são transmitidas.

Quando uma mensagem é colocada em uma fila remota, o Advanced Message Security intercepta a operação e processa a mensagem de acordo com a política configurada para a fila remota. Por exemplo, para uma política de criptografia a mensagem é criptografada antes de ser transmitida para que o IBM MQ a manuseie. Depois que Advanced Message Security tiver processado a mensagem colocada em uma fila remota, o IBM MQ a colocará em uma fila de transmissão associada e a redirecionará para o gerenciador de filas de destino e para a fila de destino.

Quando uma operação GET for executada na fila local, o Advanced Message Security tentará decodificar a mensagem de acordo com a política configurada na fila local. Para que a operação seja bem-sucedida, a política usada para decriptografar a mensagem deve ser idêntica à que foi usada para criptografá-la. Qualquer discrepância fará com que a mensagem seja rejeitada.

Se, por algum motivo ambas as políticas não puderem ser configuradas ao mesmo tempo, um suporte ao lançamento em estágios será fornecido. A política pode ser configurada em uma fila local com o sinalizador de tolerância ligado, o que indica que uma política associada a uma fila pode ser ignorada quando uma tentativa de recuperar uma mensagem da fila envolve uma mensagem que não possui o conjunto de política de segurança. Neste caso, GET tentará decriptografar a mensagem, mas permitirá que as mensagens não criptografadas sejam entregues. Dessa maneira as políticas em filas remotas podem ser configuradas após a filas locais terem sido protegidas (e testadas).

Não se esqueça: Remova o sinalizador de tolerância quando a apresentação do Advanced Message Security for concluída.

Referências relacionadas

setmqspl (configurar política de segurança)

Roteando mensagens protegidas com o AMS usando o IBM Integration Bus

O Advanced Message Security pode proteger mensagens em uma infraestrutura na qual o IBM Integration Bus ou WebSphere Message Broker 8.0.0.1 (ou mais recente) é instalado. É necessário entender a natureza de ambos os produtos antes de aplicar a segurança no ambiente do IBM Integration Bus.

Sobre esta tarefa

O Advanced Message Security fornece segurança de ponta a ponta da carga útil da mensagem. Isso significa que apenas as partes especificadas como emissores e destinatários válidos de uma mensagem são capazes de produzir ou recebê-la. Isto significa que, a fim de proteger as mensagens que fluem por meio do IBM Integration Bus, é possível permitir que o IBM Integration Bus processe mensagens sem saber seu conteúdo (Cenário 1) ou torná-lo um usuário autorizado capaz de receber e enviar mensagens (Cenário 2).

Cenário 1 - O Integration Bus não pode ver o conteúdo da mensagem

Antes de começar

É necessário ter seu IBM Integration Bus conectado a um gerenciador de filas existente. Substitua *QMgrName* com esse nome do gerenciador de filas existente nos comandos a seguir.

Sobre esta tarefa

Neste cenário, Alice coloca uma mensagem protegida em uma fila de entrada `QIN`. Com base na propriedade da mensagem `routeTo`, a mensagem é roteada para *bob's* (`QBOB`),[1](`QCECIL`), ou a fila padrão (`QDEF`). O roteamento é possível porque o Advanced Message Security protege apenas a carga útil da mensagem e não seus cabeçalhos e propriedades, que permanecem desprotegidos e podem ser lidos pelo IBM Integration Bus. O Advanced Message Security é usado somente por *alice*, *bob* e *cecil*. Não é necessário instalar ou configurar para o IBM Integration Bus.

O IBM Integration Bus recebe a mensagem protegida da fila de alias não protegido para evitar qualquer tentativa de decriptografar a mensagem. Se fosse usar a fila protegida diretamente a mensagem seria colocada na fila DEAD LETTER como impossível de decriptografar. A mensagem é roteada pelo IBM Integration Bus e chega na fila de destino inalterada. Portanto, ela continua assinada pelo autor original (ambos *bob* e *cecil* aceitam somente mensagens enviadas por *alice*) e protegida como antes (somente *bob* e *cecil* pode lê-la). O IBM Integration Bus coloca a mensagem roteada em um alias desprotegido. Os destinatários recuperam a mensagem a partir de uma fila de saída protegida em que AMS decriptografará a mensagem de modo transparente.

Procedimento

1. Configure *alice*, *bob* e *cecil* para usar o Advanced Message Security conforme descrito no **Guia de iniciação rápida** (Windows ou AIX).

   Assegure-se de que as etapas a seguir sejam concluídas:

   - Criando e autorizando usuários
   - Criando banco de dados de chaves e certificados
   - Criando keystore.conf
2. Forneça o certificado da *alice* para *bob* e *cecil* para que *alice* possa ser identificada por eles ao verificarem assinaturas digitais em mensagens.

[1] cecil's

Faça isso extraindo o certificado que identifica *alice* em um arquivo externo e, em seguida, incluindo o certificado extraído nos keystores *do bob* e *da cecil*. É importante que você use o método descrito na **Tarefa 5. Compartilhando certificados** no **Guia de Iniciação Rápida** (Windows ou AIX).

3. Forneça os certificados de *bob* e *do cecil* para *alice*, para que *alice* possa enviar mensagens criptografadas para *bob* e *cecil*.

   Faça isso usando o método especificado na etapa anterior.

4. Em seu gerenciador de filas, defina as filas locais chamadas `QIN`, `QBOB`, `QCECIL` e `QDEF`.

```
DEFINE QLOCAL(QIN)
```

5. Configure a política de segurança para a fila `QIN` para uma configuração elegível. Use a configuração idêntica para as filas `QBOB`, `QCECIL` e `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"
```

   Este cenário supõe a política de segurança na qual *alice* é o único emissor autorizado e *bob* e *cecil* são os destinatários.

6. Defina as filas de alias `AIN`, `ABOB` e `ACECIL` fazendo referência à filas locais `QIN`, `QBOB` e `QCECIL` respectivamente.

```
DEFINE QALIAS(AIN) TARGET(QIN)
```

7. Verifique se a configuração de segurança para o alias especificado na etapa anterior não está presente; caso contrário, defina sua política para NONE.

```
dspmqspl -m QMgrName -p AIN
```

8. No IBM Integration Bus crie um fluxo de mensagens para rotear as mensagens que chegam na fila de alias `AIN` para os nós BOB, CECIL ou DEF, dependendo da propriedade `routeTo` da mensagem. Para fazê-lo:
   a) Crie um nó `MQInput` chamado `IN` e designe o alias `AIN` como seu nome da fila.
   b) Crie nós `MQOutput` chamados `BOB`, `CECIL` e `DEF` e designe filas de alias `ABOB`, `ACECIL` e `ADEF` como seus respectivos nomes de filas.
   c) Crie um nó de rota e chame-o TEST.
   d) Conecte o nó `IN` ao terminal de entrada do nó TEST.
   e) Crie os terminais de saída `bob` e `cecil` para o nó TEST.
   f) Conecte o terminal de saída `bob` ao nó `BOB`.
   g) Conecte o terminal de saída `cecil` ao nó `CECIL`.
   h) Conecte o nó DEF para o terminal de saída padrão.
   i) Aplique as regras a seguir:

```
$Root/MQRFH2/usr/routeTo/text()="bob"
```

```
$Root/MQRFH2/usr/routeTo/text()="cecil"
```

9. Implemente o fluxo de mensagens para o componente de tempo de execução do IBM Integration Bus.
10. Executando como o usuário `Alice`, coloque uma mensagem que também contém uma propriedade de mensagem chamada `routeTo` com um valor de `bob` ou `cecil`. Executar o aplicativo de amostra **amqsstm** permitirá que você faça isso.

```
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. Executando como o usuário *bob*, recupere a mensagem da fila QBOB usando o aplicativo de amostra **amqsget**.

Resultados

Quando *alice* colocar uma mensagem na fila QIN a mensagem será protegida. Ela é recuperada em forma protegida pelo IBM Integration Bus a partir da fila de alias AIN. O IBM Integration Bus decide para onde rotear a mensagem ao ler a propriedade routeTo que é, como todas as propriedades, não criptografada. O IBM Integration Bus coloca a mensagem no alias desprotegido, evitando sua proteção adicional. Quando recebida por *bob* ou *cecil* da fila, a mensagem será decriptografada e a assinatura digital será verificada.

Cenário 2 - O Integration Bus pode ver o conteúdo da mensagem

Sobre esta tarefa

Neste cenário, um grupo de indivíduos têm permissão para enviar mensagens para o IBM Integration Bus. Outro grupo está autorizado a receber mensagens que são criadas pelo IBM Integration Bus. A transmissão entre as partes e o IBM Integration Bus não pode ser espionada do tráfego de rede.

Lembre-se de que o IBM Integration Bus lê as políticas de proteção e certificados somente quando uma fila for aberta, portanto deve-se recarregar o grupo de execução após fazer quaisquer atualizações para as políticas de proteção para que as mudanças entrem em vigor.

```
mqsireload execution-group-name
```

Se o IBM Integration Bus for considerado uma parte autorizada com permissão para ler ou assinar a carga útil da mensagem, deve-se configurar o Advanced Message Security para o usuário que inicia o serviço do IBM Integration Bus. Esteja ciente de que ele não é necessariamente o mesmo usuário que coloca/obtém as mensagens nas filas nem o usuário que cria e implementa os aplicativos do IBM Integration Bus.

Procedimento

1. Configure *alice*, *bob*, *cecil* e *dave* e o usuário do serviço IBM Integration Bus para usar o Advanced Message Security conforme descrito no **Guia de iniciação rápida** (Windows ou AIX).

   Assegure-se de que as etapas a seguir sejam concluídas:

   - Criando e autorizando usuários
   - Criando banco de dados de chaves e certificados
   - Criando keystore.conf

2. Forneça os certificados de *alice*, *bob*, *cecil* e *dave* ao usuário do serviço do IBM Integration Bus.

   Faça isso extraindo cada um dos certificados que identificam *alice*, *bob*, *cecil* e *dave* em arquivos externos e, em seguida, incluindo os certificados extraídos no keystore do IBM Integration Bus. É importante que você use o método descrito na **Tarefa 5. Compartilhando certificados** no **Guia de Iniciação Rápida** (Windows ou AIX).

3. Forneça certificado de usuário de serviço do IBM Integration Bus para *alice*, *bob*, *cecil* e *dave*.

   Faça isso usando o método especificado na etapa anterior.

Nota: *alice* e *bob* precisam do certificado de usuário do serviço do IBM Integration Bus para criptografar as mensagens corretamente. O usuário do serviço do IBM Integration Bus precisa dos certificados de *alice* e *bob* para verificar os autores das mensagens. O usuário do serviço do IBM Integration Bus precisa dos certificados de *cecil* e *dave* para criptografar as mensagens para eles. *cecil* e *dave* precisam do certificado de usuário do serviço do IBM Integration Bus para verificar se a mensagem vem do IBM Integration Bus.

4. Defina uma fila local denominada `IN` e defina a política de segurança com *alice* e *bob* especificados como autores e o usuário do serviço para o IBM Integration Bus especificado como destinatário:

```
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. Defina uma fila local denominada `OUT` e defina a política de segurança com o usuário do serviço para o IBM Integration Bus especificado como autor e *cecil* e *dave* especificados como destinatários:

```
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. No IBM Integration Bus crie um fluxo de mensagens com um nó `MQInput` e `MQOutput`. Configure o nó `MQInput` para usar a fila `IN` e o nó `MQOutput` para usar a fila `OUT`.
7. Implemente o fluxo de mensagens para o componente de tempo de execução do IBM Integration Bus.
8. Executando como os usuários *alice* ou *bob*, coloque uma mensagem na fila `IN` usando o aplicativo de amostra **amqsput**.
9. Executando como os usuários *cecil* ou *dave*, recupere a mensagem da fila `OUT` usando o aplicativo de amostra **amqsget**.

Resultados

As mensagens enviadas por *alice* ou *bob* para a fila de entrada `IN` são criptografadas, permitindo que somente o IBM Integration Bus as leia. O IBM Integration Bus aceita somente mensagens de *alice* e *bob* e rejeita quaisquer outras. As mensagens aceitas são processadas adequadamente e, em seguida, assinadas e criptografadas com as chaves de *cecil* e *dave* antes de serem colocadas na fila de saída `OUT`. Somente *cecil* e *dave* são capazes de ler, as mensagens não assinadas pelo IBM Integration Bus são rejeitadas.

Usando o Advanced Message Security com o Managed File Transfer

Este cenário explica como configurar o Advanced Message Security para fornecer privacidade de mensagem para dados que estão sendo enviados por meio de um Managed File Transfer.

Antes de começar

Assegure-se de que você tenha o componente Advanced Message Security instalado na instalação do IBM MQ que hospeda as filas usadas pelo Managed File Transfer que deseja proteger.

Se seus agentes do Managed File Transfer estiverem se conectando no modo de ligação, assegure-se de que você também tenha o componente IBM Global Security Kit (GSKit) instalado em sua instalação local

Sobre esta tarefa

Quando a transferência de dados entre dois agentes do Managed File Transfer for interrompida, dados possivelmente confidenciais poderão permanecer desprotegido nas filas do IBM MQ subjacentes usadas para gerenciar a transferência. Este cenário explica como configurar e usar o Advanced Message Security para proteger tais dados no Managed File Transfer.

Neste cenário, consideramos uma topologia simples que compreende uma máquina com duas Managed File Transfer filas e dois agentes, `AGENT1` e `AGENT2`, compartilhando um único gerenciador de filas, conforme descrito no cenário Managed File Transfer cenário. Ambos os agentes se conectam da mesma maneira, no modo de ligações ou no modo cliente.

1. Criando certificados

Antes de começar

Este cenário usa um modelo simples em que um usuário `ftagent` em um grupo FTAGENTS é usado para executar os processos do Managed File Transfer Agent. Se você estiver usando os seus próprios nomes do usuário e do grupo, mude os comandos de modo correspondente.

Sobre esta tarefa

O Advanced Message Security usa a criptografia de chave pública para assinar e/ou criptografar mensagens em filas protegidas.

Nota:

- Se os seus agentes do Managed File Transfer estiverem em execução em modo de ligações, os comandos que você usar para criar um keystore do CMS (Cryptographic Message Syntax) serão detalhados no **Guia de iniciação rápida** (Windows ou AIX) para a sua plataforma.
- Se os agentes do Managed File Transfer estiverem em execução no modo cliente, os comandos que você precisará para criar um JKS (Java Keystore) são detalhadas no "Guia de iniciação rápida para o AMS com os clientes Java" na página 641.

Procedimento

1. Crie um certificado autoassinado para identificar o usuário `ftagent` conforme detalhado no Guia de iniciação rápida apropriado.
   Use um Nome distinto (DN) conforme a seguir:

   ```
   CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>
   ```

2. Crie um arquivo `keystore.conf` para identificar a localização do keystore e o certificado dentro dele conforme detalhado no Guia de iniciação rápida apropriado.

2. Configurando a proteção de mensagens

Sobre esta tarefa

É necessário definir uma política de segurança para a fila de dados usada pelo AGENT2 usando o comando **`setmqspl`**. Nesse cenário, o mesmo usuário é usado para iniciar ambos os agentes, e, portanto, o signatário e o destinatário DN são iguais e corresponderem ao certificado gerado.

Procedimento

1. Encerre os agentes do Managed File Transfer em preparação para a proteção usando o comando **`fteStopAgent`**.
2. Crie uma política de segurança para proteger a fila SYSTEM.FTE.DATA.AGENT2.

   ```
   setmqspl -m hubQM -p SYSTEM.FTE.DATA.AGENT2 -s SHA1 -a "CN=ftagent, OU=MFT,
   O=<organisation>, L=<location>, ST=<state>, C=<country>"
   -e AES128 -r "CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>"
   ```

3. Assegure-se de que o usuário que está executando o processo do Managed File Transfer Agent tenha acesso para procurar a fila de política do sistema e colocar mensagens na fila de erros.

   ```
   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. Reinicie os agentes do Managed File Transfer usando o comando **`fteStartAgent`**.

5. Confirme se seus agentes foram reiniciados com sucesso usando o comando **fteListAgents** e verifique se os agentes estão em status READY.

Resultados

Você agora é capaz de enviar transferências do AGENT1 para AGENT2 e o conteúdo do arquivo será transmitido de forma segura entre os dois agentes.

Visão Geral de Instalação do Advanced Message Security

Instale o componente Advanced Message Security em várias plataformas.

Procedimento

- Multi
  Instale o Advanced Message Security em multiplataformas.
- z/OS
  Instale o IBM MQ Advanced for z/OS.
- z/OS
  Instale o IBM MQ Advanced for z/OS Value Unit Edition.

Tarefas relacionadas

Desinstalando o Advanced Message Security

z/OS Auditing for AMS on z/OS

Advanced Message Security (AMS) for z/OS provides a means for optional auditing of operations by applications on policy protected queues. When enabled, IBM System Management Facility (SMF) audit records are generated for the success and failure of these operations on policy-protected queues. Operations audited include MQPUT, MQPUT1, and MQGET.

Auditing is disabled by default, however, you can activate auditing by configuring _AMS_SMF_TYPE and _AMS_SMF_AUDIT in the configured Language Environment® _CEE_ENVFILE file for the AMS address space. For more information, see Create procedures for Advanced Message Security. The _AMS_SMF_TYPE variable is used to designate the SMF record type and is a number between 128 and 255. A SMF record type of 180 is usual, however is not mandatory. Auditing is disabled by specifying a value of 0. The _AMS_SMF_AUDIT variable configures whether audit records are created for operations that are successful, operations that fail, or both. The auditing options can also be dynamically changed while AMS is active using operator commands. For more information, see Operating Advanced Message Security.

The SMF record is defined using subtypes, with subtype 1 being a general auditing event. The SMF record contains all data relevant to the request being processed.

The SMF record is mapped by the CSQ0KSMF macro (note the zero in the macro name), which is provided in the target library SCSQMACS. If you are writing data-reduction programs for SMF data, you can include this mapping macro to aid in the development and customization of SMF post-processing routines.

In the SMF records produced by Advanced Message Security for z/OS, the data is organized into sections. The record consists of:

- a standard SMF header
- a header extension defined by Advanced Message Security for z/OS
- a product section
- a data section

The product section of the SMF record is always present in the records produced by Advanced Message Security for z/OS. The data section varies based on subtype. Currently, one subtype is defined and therefore a single data section is used.

SMF is described in the z/OS System Management Facilities manual (SA22-7630). Valid record types are described in the SMFPRMxx member of your system PARMLIB data set. See SMF documentation for more information.

Advanced Message Security audit report generator (CSQ0USMF)

Advanced Message Security for z/OS provides an audit report generator tool called CSQ0USMF which is provided in the installation SCSQAUTH library. Sample JCL to run the CSQ0USMF utility called CSQ40RSM is provided in the installation library SCSQPROC.

Before running the CSQ0USMF utility, the SMF type 180 records must be dumped from the system SMF data sets to a sequential data set. As an example, this JCL dumps SMF type 180 records from an SMF data set, and transfers them to a target data set:

```
//IFAUDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYSn.MANn.syst,DISP=SHR
//OUTDD1 DD DSN=your.target.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INDD(INDD1,OPTIONS(DUMP))
OUTDD(OUTDD1,TYPE(180))
/*
```

You must verify the actual SMF data set names used by your installation. The target data set for the dumped records must have a record format of VBS, and a record length of 32760.

Note: If SMF logstreams are being used, you must use program IFASMFDL to dump a logstream out to a sequential dataset. See Processing type 116 SMF records for an example of the JCL used.

The target data set can then be used as input to the CSQ0USMF utility to produce an AMS audit report. For example:

```
//STEP1 EXEC PGM=CSQ0USMF,
// PARM=('/ -SMFTYPE 180 -M qmgr')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//    DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SMFIN DD DSN=your.target.dataset,DISP=SHR
//
```

The CSQ0USMF program accepts two optional parameters, which are listed in Table 103 on page 653:

Table 103. CSQ0USMF optional parameters

Parameter	Value	Description
SMFTYPE	nnn	The SMF record type applicable to the audit report. The CSQ0USMF program uses only SMF records that match the SMFTYPE value when generating the report. If you do not specify SMFTYPE, a default value of 180 is used.
M	qmgr	The IBM MQ queue manager name applicable to the audit report. If you do not specify the -M parameter, the audit report will include all audit records for all queue managers represented in the SMFIN data set.

Usando keystores e certificados com o AMS

Para fornecer proteção criptográfica transparente para aplicativos IBM MQ, o Advanced Message Security usa o arquivo keystore, em que certificados de chave pública e uma chave privada são armazenados. Ativado z/OS, um conjunto de chaves SAF é usado em vez de um arquivo keystore

No Advanced Message Security, usuários e aplicativos são representados por identidades de infraestrutura de chave pública (PKI). Esse tipo de identidade é usado para assinar e criptografar mensagens. A identidade PKI é representada pelo campo **nome distinto (DN)** do sujeito em um certificado que está associado com mensagens assinadas e criptografadas. Para um usuário ou aplicativo criptografar suas mensagens eles requerem acesso ao arquivo keystore no qual os certificados e chaves privadas e públicas associadas são armazenados.

ALW No AIX, Linux, and Windows, o local do armazenamento do keystore é fornecido no arquivo de configuração do keystore, que é `keystore.conf` por padrão. Cada usuário do Advanced Message Security deve ter o arquivo de configuração keystore que aponta para um arquivo keystore. O Advanced Message Security aceita o formato de arquivos de armazenamento de chaves a seguir: `.kdb`, `.jceks`, `.jks`.

O local padrão do arquivo `keystore.conf` é:

- Linux IBM i AIX No IBM i, AIX and Linux: `$HOME/.mqs/keystore.conf`
- Windows No Windows: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

Se você estiver usando um nome do arquivo keystore e um local especificados, deverá especificar isso com a variável de ambiente **MQS_KEYSTORE_CONF** , conforme mostrado nos comandos de exemplo a seguir:

- Para Java: `java -D`*`MQS_KEYSTORE_CONF=path/filename app_name`*
- Para um cliente e servidor C:
  - Linux AIX No AIX and Linux: `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - Windows No Windows: `set MQS_KEYSTORE_CONF=`*`path\filename`*

    Nota: O caminho em Windows pode e deve especificar a letra da unidade se mais de uma letra de unidade estiver disponível.

Protegendo informações confidenciais no arquivo `keystore.conf`

Para acessar informações confidenciais do arquivo keystore, como senhas, deve-se fornecer tokens para que o IBM MQ Advanced Message Security (AMS) possa acessar o keystore e assinar e criptografar mensagens.

É necessário proteger as informações confidenciais contidas no arquivo de configuração do keystore usando o comando **runamscred** fornecido com AMS. Consulte “Configurando a proteção de senha do AMS para arquivos de configuração” na página 673 para obter detalhes sobre como proteger arquivos de configuração.

Ao proteger senhas, é necessário usar uma chave de criptografia customizada e forte. Para acessar as senhas durante o tempo de execução, essa chave de criptografia deve ser fornecida para AMS.

Existem dois métodos para fornecer o local do arquivo-chave de criptografia, que são:

- propriedade de configuração **amscred.keyfile** no arquivo `keystore.conf`
- **MQS_AMSCRED_KEYFILE** variável de ambiente

A ordem de precedência é **MQS_AMSCRED_KEYFILE**, seguida por **amscred.keyfile**, e depois a chave padrão.

Conceitos relacionados

“Nomes distintos do remetente no AMS” na página 682

Os nomes distintos (DNs) do emissor identificam usuários que estão autorizados a colocar mensagens em uma fila. Um emissor usa seu certificado para assinar uma mensagem antes de colocar a mensagem em uma fila.

"Nomes distintos do destinatário no AMS" na página 684
Os nomes distintos (DN) do destinatário identificam usuários que estão autorizados a recuperar mensagens em uma fila.

Estrutura do arquivo de configuração do keystore (keystore.conf) para AMS

O arquivo de configuração do keystore (`keystore.conf`) aponta Advanced Message Security para o local do keystore apropriado.

Cada um dos seguintes tipos de arquivo de configuração tem um prefixo:

AMSCRED
Os parâmetros que se relacionam com o sistema de proteção de senha.

CMS
Certificate Management System, as entradas de configuração são prefixadas com: `cms.`

PKCS#11
Public Key Cryptography Standard #11, entradas de configuração são prefixadas com: `pkcs11.`

IBM i **PEM**
Formato Privacy Enhanced Mail, entradas de configuração são prefixadas com: `pem.`

JKS
Java KeyStore, entradas de configuração são prefixadas com: `jks.`

JCEKS
Java Cryptographic Encryption KeyStore, entradas de configuração são prefixadas com: `jceks.`

z/OS MQ Adv. VUE **JCERACFKS**
Java Cryptographic Encryption RACF keyring KeyStore, entradas de configuração são prefixadas com: `jceracfks`.

Importante: A partir do IBM MQ 9.0, os valores `JCEKS.provider` e `JKS.provider` são ignorados. O provedor Bouncy Castle é usado, junto com qualquer provisão de JCE/JCE fornecida pelo JRE em uso. Para obter mais informações, consulte "Suporte para JREs não IBM com o AMS" na página 659.

Exemplo de estruturas para keystores:

CMS

```
cms.keystore = /dir/keystore_file
cms.certificate = certificate_label
```

PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.encrypted = no
```

IBM i PEM

```
pem.private = /dir/keystore_file_private_key
pem.public = /dir/keystore_file_public_keys
pem.password = password
pem.encrypted = no
```

JavaJKS

```
jks.keystore = dir/Keystore
jks.certificate = certificate_label
jks.encrypted = no
jks.keystore_pass = password
jks.key_pass = password
```

JavaJCEKS

```
jceks.keystore = dir/Keystore
jceks.certificate = certificate_label
jceks.encrypted = no
jceks.keystore_pass = password
jceks.key_pass = password
```

Java JCERACFKS

```
jceracfks.keystore = safkeyring://user/keyring
jceracfks.certificate = certificate_label
```

Java PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
pkcs11.secondary_keystore_pass = password
pkcs11.encrypted = no
```

Tabela 104. Resumo dos parâmetros necessários para cada tipo de arquivo de configuração

Parâmetros	Necessário	Tipo de arquivo de configuração				
		Java (PKCS#11, JKS, JCEKS e JCERACFKS)	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
keystore	✓	✓			✓	
IBM i private	✓		IBM i ✓			
IBM i public	✓		IBM i ✓			
IBM i password	✓		IBM i ✓			
library	✓	✓		✓		
certificate	✓	✓		✓	✓	
token	✓	✓		✓		
token_pin	✓	✓		✓		

Tabela 104. Resumo dos parâmetros necessários para cada tipo de arquivo de configuração (continuação)

Parâmetros	Necessário	Tipo de arquivo de configuração				
		Java (PKCS#11, JKS, JCEKS e JCERACFKS)	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
`secondary_keystore`	✓	✓		✓		
`secondary_keystore_password`	✓	✓				
`encrypted`		✓	IBM i ✓	✓		
`keystore_pass`	✓	✓				
`key_pass`		✓				
`provider`		✓				
`keyfile`						✓ Você

Observe que é possível incluir comentários usando o símbolo #

Os parâmetros de arquivo de configuração são definidos como seguir:

keystore
Somente configuração do CMS e Java.

Caminho para o arquivo keystore para configuração de CMS, JKS e JCEKS.

z/OS MQ Adv. VUE URI para o conjunto de chaves RACF para configuração JCERACFKS.

Importante:

- O caminho para o arquivo keystore não deve incluir a extensão do arquivo.
- z/OS MQ Adv. VUE O URI para o conjunto de chaves RACF deve estar no formato:

  ```
  safkeyring://user/keyring
  ```

  em que:
  - *user* é o ID do usuário que possui o conjunto de chaves
  - *keyring* é o nome do conjunto de chaves.

IBM i **private**
Somente configuração do PEM.

O nome do arquivo de um arquivo que contém a chave privada e o certificado no formato PEM.

IBM i **public**
Somente configuração do PEM.

O nome do arquivo de um arquivo que contém certificados públicos confiáveis no formato PEM.

IBM i **password**

Somente configuração do PEM.

Senha usada para decriptografar uma chave privada criptografada.

É necessário proteger esse campo usando a ferramenta de proteção de senha nativa do AMS; consulte “Protegendo senhas” na página 659

library

Somente PKCS#11.

Nome do caminho da biblioteca PKCS#11.

certificate

CMS, PKCS#11 e somente configuração do Java.

Etiqueta do certificado.

token

Somente PKCS#11.

Rótulo do token.

token_pin

Somente PKCS#11.

PIN para desbloquear o token.

Somente para operações Java; é necessário proteger esse campo usando a ferramenta de proteção de senha Java AMS. Consulte “Protegendo senhas” na página 659.

Apenas para operações nativas, é necessário proteger esse campo usando a ferramenta de proteção de senha nativa do AMS; consulte “Protegendo senhas” na página 659.

secondary_keystore

Somente PKCS#11.

Nome do caminho do keystore CMS, fornecido sem a extensão `.kdb`, que contém certificados âncora (certificados raiz) requeridos por certificados armazenados no token PKCS #11. O keystore secundário também pode conter os certificados intermediários na cadeia de confiança, bem como certificados do destinatário definidos na política de segurança de privacidade. Esse keystore CMS deve ser acompanhado por um arquivo stash que deve estar localizado no mesmo diretório que o keystore secundário.

Para ambientes Java, é necessário um keystore JKS e deve-se fornecer um **secondary_keystore_password**.

secondary_keystore_password

Somente Java PKCS#11.

A senha para o keystore JKS fornecido por meio da propriedade `secondary_keystore`. É necessário proteger esse campo usando a ferramenta de proteção de senha do Java AMS; consulte “Protegendo senhas” na página 659.

encrypted

Java e, de IBM MQ 9.3.0, PKCS#11 e IBM i PEM apenas.

Status da senha.

keystore_pass

Somente configuração do Java.

Senha do arquivo de armazenamento de chaves.

Somente para operações Java. É necessário proteger esse campo usando a ferramenta de proteção de senha do Java AMS; consulte “Protegendo senhas” na página 659.

key_pass

Somente configuração do Java.

A senha para a chave privada do usuário.

Somente para operações Java; é necessário proteger esse campo usando a ferramenta de proteção de senha Java AMS. Consulte “Protegendo senhas” na página 659.

keyfile
Fornece o local da chave inicial a ser usado ao proteger ou decriptografar senhas contidas neste arquivo de configuração. Consulte “Protegendo senhas” na página 659

provider
Somente configuração do Java.

O provedor de segurança do Java que implementa os algoritmos criptográficos requeridos pelo certificado keystore.

Importante: As informações armazenadas no keystore são cruciais para o fluxo seguro de dados enviado usando o IBM MQ. Os administradores de segurança deverão prestar atenção especial quando estiverem designando permissões de arquivo para esses arquivos.

Protegendo senhas

É necessário proteger as senhas e outras informações confidenciais contidas no arquivo `keystore.conf`. Para obter mais informações, consulte **runamscred**.

Exemplo do arquivo `keystore.conf`

```
# Native AMS application configuration
cms.keystore = c:\Documents and Settings\Alice\AliceKeystore
cms.certificate = AliceCert

# Java AMS application configuration
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

Tarefas relacionadas

“Configurando a proteção de senha do AMS para arquivos de configuração” na página 673
Armazenar keystores e senhas de chaves privadas como texto sem formatação representa um risco de segurança, portanto o Advanced Message Security fornece uma ferramenta que pode misturar essas senhas usando uma chave do usuário.

Suporte para JREs não IBM com o AMS

O IBM MQ classes for Java e o IBM MQ classes for JMS suportam a operação Advanced Message Security ao executar com JREs não IBM.

O Advanced Message Security (AMS) implementa o Cryptographic Message Syntax (CMS). A sintaxe CMS é usada para assinar digitalmente, compilar, autenticar ou criptografar conteúdo de mensagem arbitrário.

No IBM MQ 9.0, o suporte do Advanced Message Security no IBM MQ classes for Java e IBM MQ classes for JMS usa os pacotes Bouncy Castle de software livre para suportar o CMS. Isso significa que essas classes podem suportar a operação Advanced Message Security ao executar com JREs não IBM.

Antes da IBM MQ 9.0, o Advanced Message Security não era suportado em JREs não IBM em clientes Java. O suporte do Advanced Message Security no IBM MQ classes for Java e IBM MQ classes for JMS dependia do suporte do CMS fornecido especificamente pela implementação da IBM do Java Cryptography Extensions (JCE). Por causa dessa restrição, a funcionalidade estava disponível somente ao usar um Java runtime environment (JRE) que incluía o provedor JCE do Java.

Local e numeração de versão para arquivos JAR do Bouncy Castle

Os arquivos JAR do Bouncy Castle que são necessários para suporte para JREs não IBM são incluídos como parte do pacote de instalação do IBM MQ classes for Java e do IBM MQ classes for JMS.

Os arquivos JAR do Bouncy Castle usados são os arquivos a seguir:

O arquivo JAR do provedor, que é fundamental para as operações do Bouncy Castle.

V 9.4.0 Em IBM MQ 9.4.0, esse arquivo JAR é chamado `bcprov-jdk18on.jar`.

O arquivo JAR "PKIX", que contém o suporte para operações CMS que são usadas pelo Advanced Message Security.

V 9.4.0 Em IBM MQ 9.4.0, esse arquivo JAR é chamado `bcpkix-jdk18on.jar`.

O arquivo JAR "util", que contém as classes usadas pelos outros arquivos jar do Bouncy Castle.

V 9.4.0 Em IBM MQ 9.4.0, esse arquivo JAR é chamado `bcutil-jdk18on.jar`.

Dependências

A IBM MQ 9.1 e classes mais recentes foram testadas com JREs do IBM e com JREs do Oracle. Eles também devem ser executados com sucesso sob qualquer JRE compatível com J2SE. Entretanto, é necessário observar as dependências a seguir:

- Não há mudanças na configuração do Advanced Message Security.
- As classes Bouncy Castle são usadas somente para operações CMS. Todas as outras operações relacionadas à segurança, por exemplo, o acesso de keystore de exemplo, a criptografia real de dados e o cálculo de somas de verificação de assinatura, usam a funcionalidade fornecida pelo JRE.

  Importante: Por este motivo, o JRE usado deve incluir uma implementação de provedor JCE.
- Para usar alguns algoritmos de criptografia *avançada*, pode ser necessário instalar os arquivos de políticas *sem restrições* para a implementação JCE do JRE.

  Consulte a documentação do JRE para obter mais detalhes.
- Se você tiver ativado a segurança do Java:
  - Inclua `java.security.SecurityPermissioninsertProvider.BC` no aplicativo para que as classes do Bouncy Castle possam ser usadas como um provedor de segurança.
  - Conceda `java.security.AllPermission` aos arquivos JAR do Bouncy Castle.

    V 9.4.0 Em IBM MQ 9.4.0, esses arquivos são:

    ```
    mq_install_dir/java/lib/bcutil-jdk18on.jar
    mq_install_dir/java/lib/bcpkix-jdk18on.jar
    mq_install_dir/java/lib/bcprov-jdk18on.jar
    ```

Conceitos relacionados

O que é instalado para classes do IBM MQ para JMS

O que é instalado para classes do IBM MQ para Java

Multi Intercepção do Agente do Canal de Mensagens (MCA) e AMS

A intercepção de MCA permite que um gerenciador de filas em execução sob o IBM MQ ative seletivamente as políticas a serem aplicadas para os canais de conexão do servidor.

A intercepção de MCA permite aos clientes que permanecem fora do AMS continuem conectados a um gerenciador de filas e suas mensagens sejam criptografadas e decriptografadas.

A intercepção de MCA destina-se a fornecer a capacidade do AMS quando o AMS não puder ser ativado no cliente. Observe que usar a intercepção de MCA e um cliente ativado por AMS leva à dupla proteção de mensagens que pode ser problemática para aplicativos de recebimento. Para obter mais informações, consulte “Desativando o Advanced Message Security no cliente” na página 663.

Nota: Os interceptores de MCA não são suportados para canais AMQP ou MQTT.

Arquivo de configuração do keystore

Por padrão, o arquivo de configuração keystore para intercepção de MCA é `keystore.conf` e está localizado no diretório `.mqs` no caminho do diretório HOME do usuário que iniciou o gerenciador de filas ou o listener. O keystore também pode ser configurado usando a variável de ambiente MQS_KEYSTORE_CONF. Para obter mais informações sobre como configurar o keystore do AMS, veja "Usando keystores e certificados com o AMS" na página 654.

Para ativar a intercepção MCA, deve-se fornecer o nome de um canal que você deseja usar no arquivo de configuração keystore. Para a intercepção de MCA, somente um tipo de keystore cms pode ser usado.

Veja "Exemplo de intercepção do MCA para o AMS" na página 661 para obter um exemplo de como configurar a intercepção de MCA.

Atenção: Deve-se concluir a autenticação de cliente e a criptografia nos canais selecionados, por exemplo, usando SSL e SSLPEER ou CHLAUTH TYPE(SSLPEERMAP), para assegurar que somente os clientes autorizados possam se conectar e usar esse recurso.

IBM i

Se sua empresa usar o IBM ie você tiver selecionado uma autoridade de certificação (CA) comercial para assinar seu certificado, o Digital Certificate Manager criará uma solicitação de certificado no formato PEM (Privacy-Enhanced Mail). Deve-se encaminhar a solicitação para a CA escolhida.

Para fazer isso, deve-se usar o comando a seguir para selecionar o certificado correto para o canal especificado em `channelname`:

```
pem.certificate.channel.channelname
```

Exemplo de intercepção do MCA para o AMS

Uma tarefa de exemplo de como configurar uma intercepção de MCA do AMS.

Antes de começar

Atenção: Deve-se concluir a autenticação de cliente e a criptografia nos canais selecionados, por exemplo, usando SSL e SSLPEER ou CHLAUTH TYPE(SSLPEERMAP), para assegurar que somente os clientes autorizados possam se conectar e usar esse recurso.

Se sua empresa usar o IBM ie você tiver selecionado uma autoridade de certificação (CA) comercial para assinar seu certificado, o Digital Certificate Manager criará uma solicitação de certificado no formato PEM (Privacy-Enhanced Mail). Deve-se encaminhar a solicitação para a CA escolhida.

Sobre esta tarefa

Esta tarefa leva você através do processo de configuração do seu sistema para usar a intercepção de MCA e, em seguida, verificar a configuração.

Nota: IBM MQ, inclui os interceptores AMS e os ativa dinamicamente nos ambientes de tempo de execução do cliente e do servidor MQ .

Atenção:

- Substitua `userID` no código com seu ID do usuário.
- O procedimento a seguir não funciona conforme o esperado em IBM MQ , a menos que a intercepção do AMS esteja desativada no cliente

Procedimento

1. Crie o banco de dados de chave e certificados usando os seguintes a seguir para criar um shell script.

Além disso, mude o **INSTLOC** e **KEYSTORELOC** ou executar os comandos necessários. Observe que pode não ser necessário para o certificado para o bob.

```
INSTLOC=/opt/mqm
KEYSTORELOC=/home/userID/var/mqm
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

runmqakm -keydb -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd -stash
runmqakm -keydb -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd -stash
runmqakm -cert -create -db $KEYSTORELOC/alicekey.kdb -pw passw0rd \
-label alice_cert -dn "cn=alice,O=IBM,c=IN" -default_cert yes
runmqakm -cert -create -db $KEYSTORELOC/bobkey.kdb -pw passw0rd \
-label bob_cert  -dn "cn=bob,O=IBM,c=IN"  -default_cert yes
```

2. Compartilhe os certificados entre os dois bancos de dados de chaves para que cada usuário possa identificar com sucesso o outro.

   É importante usar o método descrito para compartilhar certificados no *Guia de Iniciação Rápida*, para a plataforma que sua empresa usa:

   Windows
   Certificados de compartilhamento da tarefa 5

   AIX and Linux
   Certificados de compartilhamento da tarefa 5

   Java clientes
   Certificados de compartilhamento da tarefa 5
3. Crie `keystore.conf` com a configuração a seguir: `Keystore.conf location: /home/userID/ssl/ams1/`

```
cms.keystore = /home/userID/ssl/ams1/alicekey
cms.certificate.channel.SYSTEM.DEF.SVRCONN = alice_cert
```

Atenção:

   a. O keystore deve estar no sistema no qual o gerenciador de filas está.
   b. Deve-se especificar um canal específico para o `cms.certificate` para ativar a intervenção do MCA e, em seguida, o gerenciador de filas executa operações do AMS em aplicativos que se conectam por meio desse canal para filas com políticas configuradas
4. Crie e inicie o gerenciador de filas AMSQMGR1
5. Defina um listener TCP usando um número de porta disponível sob controle QMGR.

   Por exemplo:

```
DEFINE LISTENER(MY.LISTENER) TRPTYPE(TCP) PORT(14567) CONTROL(QMGR)
```

6. Inicie o listener e verifique se foi iniciado corretamente.

   Por exemplo:

```
START  LISTENER(MY.LISTENER)
DISPLAY LSSTATUS(MY.LISTENER) PORT
```

7. Parar o gerenciador de fila.
8. Configure o keystore:

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. Inicie o gerenciador de fila no mesmo shell, para que a variável de ambiente MQS_KEYSTORE_CONF esteja disponível para o gerenciador de filas.
10. Configure a política de segurança e verifique:

```
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
```

Consulte setmqspl e dspmqspl para obter mais informações.

11. Configure a variável de ambiente *MQSERVER* :

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. Remova a política de segurança e verifique o resultado:

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

13. Procure a fila por meio de sua instalação do IBM MQ 9.4:

```
/opt/mq93/samp/bin/amqsbcg TESTQ AMSQMGR1
```

A saída de procura mostra as mensagens no formato criptografado.

14. Configure a política de segurança e verifique o resultado:

```
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
```

15. Execute **amqsgetc** por meio de sua instalação do IBM MQ 9.4:

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

Conceitos relacionados

"Estrutura do arquivo de configuração do keystore (keystore.conf) para AMS" na página 655
O arquivo de configuração do keystore (`keystore.conf`) aponta Advanced Message Security para o local do keystore apropriado.

Referências relacionadas

"Limitações conhecidas de AMS" na página 612
Há várias opções do IBM MQ que não são suportadas ou têm limitações para o Advanced Message Security.

Desativando o Advanced Message Security no cliente

Será necessário desativar IBM MQ Advanced Message Security (AMS) se você estiver usando um cliente IBM MQ para se conectar a um gerenciador de filas de uma versão anterior do produto e um erro 2085 (`MQRC_UNKNOWN_OBJECT_NAME`) for relatado.

Sobre esta tarefa

IBM MQ Advanced Message Security (AMS) é ativado automaticamente em um cliente IBM MQ e, portanto, por padrão, o cliente tenta verificar as políticas de segurança para objetos no gerenciador de filas.

Se esse erro for relatado, quando você estiver tentando se conectar a um gerenciador de filas por meio de uma versão anterior do produto, será possível desativar o AMS da seguinte forma:

- Para clientes Java, de qualquer uma das maneiras a seguir:
  - Ao configurar uma variável de ambiente **AMQ_DISABLE_CLIENT_AMS**
  - Configurando a propriedade do sistema Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS.
  - Usando a propriedade **DisableClientAMS** , na sub-rotina Segurança no arquivo `mqclient.ini` .
- Para clientes C, configurando uma variável de ambiente **MQS_DISABLE_ALL_INTERCEPT**.

Nota: Não é possível usar a variável de ambiente **AMQ_DISABLE_CLIENT_AMS** para clientes C. Em vez disso, é preciso usar a variável de ambiente **MQS_DISABLE_ALL_INTERCEPT** .

Procedimento

- Para desativar o AMS no cliente, use uma das opções a seguir:

  Variável de ambiente AMQ_DISABLE_CLIENT_AMS
  É necessário configurar essa variável nos seguintes casos:

  – Se estiver usando um Java runtime environment (JRE) diferente do IBM Java runtime environment (JRE)

  – Se estiver usando um cliente IBM MQ IBM MQ classes for JMS ou IBM MQ classes for Java .

  Crie a variável de ambiente **AMQ_DISABLE_CLIENT_AMS** e configure-a como TRUE no ambiente no qual o aplicativo está em execução.. Por exemplo:

  ```
  export AMQ_DISABLE_CLIENT_AMS=TRUE
  ```

  Propriedade do sistema Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS
  Para os clientes IBM MQ classes for JMS e IBM MQ classes for Java, é possível configurar a propriedade do sistema Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS para o valor TRUE para o aplicativo Java.

  Por exemplo, é possível configurar a propriedade do sistema Java como uma opção `-D` quando o comando Java é chamado:

  ```
  JM 3.0 java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/
  java/lib/com.ibm.mq.jakarta.client.jar  my.java.applicationClass
  ```

  ```
  JMS 2.0 java -Dcom.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS=TRUE -cp <MQ_INSTALLATION_PATH>/
  java/lib/com.ibm.mq.allclient.jar  my.java.applicationClass
  ```

  Alternativamente, especifique a propriedade do sistema Java dentro de um arquivo de configuração do JMS, `jms.config`, se o aplicativo utilizar este arquivo.

  Variável de ambiente MQS_DISABLE_ALL_INTERCEPT
  É necessário configurar essa variável de ambiente se você estiver usando IBM MQ com clientes nativos e for necessário desativar AMS no cliente.

  Crie a variável de ambiente **MQS_DISABLE_ALL_INTERCEPT** e configure-a como TRUE no ambiente onde o cliente está em execução. Por exemplo:

  ```
  export MQS_DISABLE_ALL_INTERCEPT =TRUE
  ```

  É possível usar a variável de ambiente **MQS_DISABLE_ALL_INTERCEPT** apenas para clientes C. Para clientes Java , é necessário usar a variável de ambiente **AMQ_DISABLE_CLIENT_AMS** no lugar.

  Propriedade DisableClientAMS no arquivo mqclient.ini
  É possível usar essa opção para clientes IBM MQ classes for JMS e IBM MQ classes for Java e para clientes C.

  Inclua o nome da propriedade `DisableClientAMS` sob sub-rotina **Security** no arquivo `mqclient.ini`, conforme mostrado no exemplo a seguir:

  ```
  Security:
  DisableClientAMS=Yes
  ```

  Também é possível ativar o AMS, conforme mostrado no exemplo a seguir:

  ```
  Security:
  DisableClientAMS=No
  ```

Como proceder a seguir

Para obter mais informações sobre problemas com a abertura de filas protegidas do AMS, consulte Problemas na abertura de filas protegidas ao usar o AMS com o JMS.

Conceitos relacionados

"Intercepção do Agente do Canal de Mensagens (MCA) e AMS" na página 660
A intercepção de MCA permite que um gerenciador de filas em execução sob o IBM MQ ative seletivamente as políticas a serem aplicadas para os canais de conexão do servidor.

Tarefas relacionadas

IBM MQ MQI client arquivo de configuração, `mqclient.ini`

Referências relacionadas

O arquivo de configuração do IBM MQ classes for JMS

Requisitos de certificado para o AMS

Os certificados devem ter uma chave pública do RSA para que sejam usados com o Advanced Message Security.

Para obter mais informações sobre os diferentes tipos de chave pública e como criá-los, consulte "Certificados digitais e compatibilidade de CipherSpec no IBM MQ" na página 48.

Extensões do uso da chave

As extensões de uso da chave colocam restrições adicionais na forma que um certificado pode ser usado.

No Advanced Message Security, o uso da chave dos certificados X.509 v3 deve ser configurado de acordo com a especificação RFC 5280.

Para a qualidade da integridade da proteção, se as extensões de uso da chave de certificado estiverem configuradas, essa configuração deverá incluir pelo menos um dos dois:

- **`nonRepudiation`**
- **`digitalSignature`**

Para a qualidade de privacidade de proteção, se as extensões de uso da chave de certificado forem configuradas, esse conjunto deverá incluir:

- **`keyEncipherment`**

Para a qualidade de confidencialidade de proteção, se as extensões de uso de chave de certificado forem configuradas, esse conjunto deverá incluir:

- **`dataEncipherment`**

O uso estendido de chaves refina ainda mais as extensões de uso das chaves. Para todas as qualidades de proteção, se o uso estendido da chave do certificado estiver configurado, a configuração deverá incluir:

- **`emailProtection`**

Conceitos relacionados

"Qualidade de proteção no AMS" na página 685
A proteção de políticas do Advanced Message Security implicam uma qualidade de proteção (QOP).

métodos de validação de certificado no AMS

É possível usar o Advanced Message Security para detectar e rejeitar os certificados revogados para que as mensagens em suas filas não sejam protegidas usando certificados que não cumprem as normas de segurança.

O AMS permite verificar a validade do certificado usando o Online Certificate Status Protocol (OCSP) ou a lista de revogação de certificados (CRL).

O AMS pode ser configurado para a verificação de OCSP ou CRL ou ambos. Se ambos os métodos forem ativados, então, por motivos de desempenho, o AMS usará o OCSP para o status de revogação primeiro. Se o status da revogação de um certificado for indeterminado após a verificação de OCSP, o AMS usará a verificação de CRL.

Observe que as verificações de OCSP e CRL são ativadas por padrão.

Conceitos relacionados

"Online Certificate Status Protocol (OCSP) no AMS" na página 666
O Online Certificate Status Protocol (OCSP) determina se um certificado foi revogado e, portanto, ajuda a determinar se o certificado pode ser confiável. O OCSP é ativado por padrão.

"Listas de revogação de certificado (CRLs) no AMS" na página 668
As CRLs contêm uma lista de certificados que foram marcados pela Autoridade de certificação (CA) como não mais confiável por vários motivos, por exemplo, a chave privada foi perdida ou comprometida.

Online Certificate Status Protocol (OCSP) no AMS

O Online Certificate Status Protocol (OCSP) determina se um certificado foi revogado e, portanto, ajuda a determinar se o certificado pode ser confiável. O OCSP é ativado por padrão.

O OCSP não é suportado nos sistemas IBM i.

Ativando a verificação de OCSP em interceptores nativos do Advanced Message Security
A verificação de Online Certificate Status Protocol (OCSP) no Advanced Message Security é ativada por padrão, com base nas informações dos certificados que estão sendo usados.

Procedimento

Inclua as seguintes opções no arquivo de configuração de chaves :

Nota: Todos as sub-rotinas de OCSP são opcionais e podem ser especificadas independentemente.

Opção	Descrição
`ocsp.enable=off`	Ative a verificação de OCSP se o certificado que está sendo verificada tiver uma Authority Info Access (AIA) com um método de acesso PKIX_AD_OCSP que contém uma URI na qual o OCSP está localizado. Possíveis valores: on ou `off`.
`ocsp.url=`*`responder_URL`*	O endereço da URL do respondente do OCSP. Se esta opção for omitida, então, a verificação de OCSP não AIA será desativada.
`ocsp.http.proxy.host=`*`OCSP_proxy`*	O endereço da URL do servidor proxy do OCSP. Se esta opção for omitida, um proxy não será usado para verificações de certificado não AIA on-line.
`ocsp.http.proxy.port=`*`port_number`*	Número da porta do servidor proxy do OCSP Se essa opção for omitida, a porta padrão 8080 será usada.
`ocsp.nonce.generation=on/off`	Gere nonce ao consultar o OCSP. O valor padrão é `off`.
`ocsp.nonce.check=on/off`	Verifique o nonce após receber a resposta do OCSP. O valor padrão é `off`.
`ocsp.nonce.size=8`	Tamanho do nonce em bytes.

Opção	Descrição
`ocsp.http.get=on/off`	Especifique HTTP GET como seu método de solicitação. Se essa opção estiver configurada como `off`, HTTP POST é utilizado. O valor padrão é `off`.
`ocsp.max_response_size=20480`	Tamanho máximo de resposta do respondente do OCSP fornecido em bytes.
`ocsp.cache_size=100`	Ative cache de resposta do OCSP interno e configure o limite para o número de entradas de cache.
`ocsp.timeout=30`	Tempo de espera por uma resposta do servidor, em segundos, após o qual o Advanced Message Security atinge o tempo limite.
`ocsp.unknown=ACCEPT`	Define o comportamento quando um servidor OCSP não pode ser alcançado dentro de um período de tempo limite. Valores possíveis: • `ACCEPT` Permite o certificado • `WARN` Permite o certificado e registra um aviso • `REJECT` Evita o certificado que está sendo usado e registra um erro

Ativando a verificação de OCSP no Java no AMS
Para ativar a verificação do OCSP para o Java no Advanced Message Security, modifique o arquivo `java.security` ou o arquivo de configuração de keystore.

Sobre esta tarefa

Existem duas maneiras de ativar a verificação de OCSP no Advanced Message Security:

Usando java.security
Verifique se o certificado contém uma extensão de certificado Authority Information Access (AIA).

Procedimento

1. Se AIA não for configurado ou você desejar substituir seu certificado, edite o arquivo `$JAVA_HOME/lib/security/java.security` com as propriedades a seguir:

   ```
   ocsp.responderURL=http://url.to.responder:port
   ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
   ```

   e ative a verificação de OCSP editando o arquivo `$JAVA_HOME/lib/security/java.security` com a linha a seguir:

   ```
   ocsp.enable=true
   ```

2. Se AIA for configurado, ative a verificação de OCSP editando o arquivo `$JAVA_HOME/lib/security/java.security` com a linha a seguir:

   ```
   ocsp.enable=true
   ```

Como proceder a seguir

Se você estiver usando o Java Security Manager to concluir a configuração, inclua a permissão Java a seguir para o `lib/security/java.policy`

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

Usando keystore.conf

Procedimento

Inclua o atributo a seguir no arquivo de configuração:

```
ocsp.enable=true
```

Importante: A configuração desse atributo no arquivo de configuração substitui as configurações java.security.

Como proceder a seguir

Para concluir a configuração, inclua as permissões Java a seguir para o `lib/security/java.policy`:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

Listas de revogação de certificado (CRLs) no AMS

As CRLs contêm uma lista de certificados que foram marcados pela Autoridade de certificação (CA) como não mais confiável por vários motivos, por exemplo, a chave privada foi perdida ou comprometida.

Para validar certificados, o Advanced Message Security constrói uma cadeia de certificados que consiste no certificado do signatário e a cadeia de certificados da autoridade de certificação (CA) até uma âncora de confiança. Uma âncora de confiança é um arquivo keystore confiável que contém um certificado confiável ou um certificado de raiz confiável que é usado para assegurar a confiança de um certificado. O AMS verifica o caminho do certificado usando um algoritmo de validação de PKIX. Quando a cadeia é criada e verificada, o AMS conclui a validação de certificado que inclui a validação do problema e a data de expiração de cada certificado na cadeia em relação à data atual, verificando se a extensão do uso da chave está presente no certificado de Entidade Final. Se a extensão for anexada ao certificado, o AMS verificará se o **digitalSignature** ou **nonRepudiation** também serão definidos. Se não forem, o `MQRC_SECURITY_ERROR` será relatado e registrado. Em seguida, o AMS faz download das CRLs dos arquivos ou do LDAP, dependendo dos valores que foram especificados no arquivo de configuração. Somente as CRLs que são codificadas no formato DER são suportadas pelo AMS. Se nenhuma configuração relacionada a CRL for localizada no arquivo de configuração do keystore, o AMS não executará nenhuma verificação de validade da CRL. Para cada certificado de autoridade de certificação o AMS consulta o LDAP para CRLs usando Nomes distintos de uma CA para localizar sua CRL. Os atributos a seguir são incluídos na consulta LDAP:

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
deltaRevocationList
deltaRevocationList;binary,
```

Nota: `deltaRevocationList` é suportado somente quando ele for especificado como pontos de distribuição.

Ativando o suporte à validação do certificado e lista de revogação de certificado em interceptores nativos

Deve-se modificar o arquivo de configuração de keystore para que o Advanced Message Security possa fazer download de CLRs do servidor Lightweight Directory Access Protocol (LDAP).

Sobre esta tarefa

IBM i Ativar o suporte de validação de certificado e lista de revogação de certificado em interceptores nativos não é suportado para o Advanced Message Security no IBM i.

Procedimento

Inclua as opções a seguir no arquivo de configuração:

Nota: Todos as sub-rotinas de CRL são opcionais e podem ser especificadas independentemente.

Opção	Descrição
`crl.ldap.host=`*`host_name`*	Nome do host do servidor LDAP.
`crl.ldap.port=`*`port_number`*	Número da porta do servidor LDAP. É possível especificar até 11 servidores. Vários hosts LDAP são usados para assegurar o failover transparente no caso de falha de conexão LDAP. É esperado que todos os servidores LDAP sejam réplicas e contenham os mesmos dados. Quando o interceptor do AMS Java se conectar com sucesso a um servidor LDAP ele não tentará fazer o download de CRLs a partir dos servidores restantes fornecidos.
`crl.cdp=off`	Use essa opção para verificar ou usar extensões CRLDistributionPoints em certificados.
`crl.ldap.version=3`	Número da versão do protocolo LDAP. Valores possíveis: 2 ou 3.
`crl.ldap.user=cn=`*`username`*	Efetue login no servidor LDAP. Se esse valor não for especificado, os atributos CRL no LDAP deverão ser legíveis no mundo
`crl.ldap.pass=`*`password`*	Senha para o servidor LDAP.
`crl.ldap.encrypted=`*`no/yes`*	Se o `clr.ldap.pass` está criptografado ou não. Consulte Protegendo senhas em arquivos de configuração do AMS para obter mais informações.
`crl.ldap.cache_lifetime=0`	Tempo de vida do cache de LDAP em segundos. Valores possíveis: 0-86400.
`crl.ldap.cache_size=50`	Tamanho do cache do LDAP. Esta opção pode ser especificada apenas se o valor `crl.ldap.cache_lifetime` for maior do que `0`.
`crl.http.proxy.host=some.host.com`	Porta do servidor proxy Http para recuperação de CRL CDP.
`crl.http.proxy.port=8080`	Número da porta do servidor proxy Http.
`crl.http.max_response_size=204800`	O tamanho máximo de CRL, em bytes, que pode ser recuperado de um servidor HTTP que é aceito pelo IBM Global Security Kit (GSKit).
`crl.http.timeout=30`	Tempo de espera por uma resposta do servidor, em segundos, após o qual o AMS atinge tempo limite.
`crl.http.cache_size=0`	Tamanho do cache HTTP, em bytes.

Opção	Descrição
`crl.unknown=ACCEPT`	Define o comportamento quando um servidor de CRL não pode ser alcançado dentro de um período de tempo limite. Valores possíveis: • `ACCEPT` Permite o certificado • `WARN` Permite o certificado e registra um aviso • `REJECT` Evita o certificado que está sendo usado e registra um erro

Ativando o suporte à lista de revogação de certificado no Java no AMS
Para ativar o suporte de CRL no Advanced Message Security, deve-se modificar o arquivo de configuração do keystore para permitir que AMS faça download de CRLs do servidor Lightweight Directory Access Protocol (LDAP) e configure o arquivo java.security.

Procedimento

1. Inclua as opções a seguir no arquivo de configuração:

Cabeçalho	Descrição
`crl.ldap.host=`*`host_name`*	Nome do host LDAP.
`crl.ldap.port=`*`port_number`*	Número da porta do servidor LDAP. É possível especificar até 11 servidores. Vários hosts LDAP são usados para assegurar o failover transparente no caso de falha de conexão LDAP. É esperado que todos os servidores LDAP sejam réplicas e contenham os mesmos dados. Quando o interceptor do AMS Java se conectar com sucesso a um servidor LDAP ele não tentará fazer o download de CRLs a partir dos servidores restantes fornecidos. O Java não usa os valores `crl.ldap.user` e crl.ldaworldp.pass. Ele não usa um usuário e senha ao se conectar a um servidor LDAP. Como consequência, os atributos de CRL LDAP devem ser legíveis mundialmente.
`crl.cdp=on/off`	Use essa opção para verificar ou usar extensões CRLDistributionPoints em certificados.

2. Modifique o arquivo `JRE/lib/security/java.security` com as propriedades a seguir:

Nome da propriedade	Descrição
`com.ibm.security.enableCRLDP`	Esta propriedade usa os valores a seguir: `true`, `false`. Se for configurada como `true`, ao fazer a verificação de revogação de certificado, as CRLs serão localizadas usando a URL da extensão de pontos de distribuição de CRL do certificado. Se for configurada como `false` ou não configurada, a verificação de CRL usando a extensão de pontos de distribuição de CRL será desativada.

Nome da propriedade	Descrição
`ibm.security.certpath.ldap.cache.lifetime`	Essa propriedade pode ser usada para configurar o tempo de vida de entradas no cache de memória de LDAP CertStore para um valor em segundos. Um valor de 0 desativa o cache; -1 significa tempo de vida ilimitado. Se não for configurado, o tempo de vida padrão será 30 segundos.
`com.ibm.security.enableAIAEXT`	Esta propriedade usa os valores a seguir: `true`, `false`. Se for configurada como `true`, quaisquer extensões de Authority Information Access que forem encontradas dentro dos certificados do caminho do certificado que está sendo construído serão examinadas para determinar se elas contêm URIs LDAP. Para cada URI LDAP localizado, um objeto LDAPCertStore será criado e incluído na coleção de CertStores que é usada para localizar outros certificados que são requeridos para construir o caminho do certificado. Se for configurada como `false` ou não configurada, objetos LDAPCertStore adicionais não serão criados.

z/OS *Ativando listas de revogação de certificado (CRLs) no z/OS*

O Advanced Message Security suporta a verificação da lista de revogação de certificado (CRL) dos certificados digitais usados para proteger os dados de mensagens

Sobre esta tarefa

Quando ativado, o Advanced Message Security irá validar certificados do destinatário quando as mensagens forem colocadas em uma fila de privacidade protegida e validar certificados do emissor quando as mensagens forem recuperadas de uma fila protegida (integridade ou privacidade). A validação nesse caso inclui a verificação de que os certificados relevantes não estão registrados em uma CRL relevante.

O Advanced Message Security usa serviços do IBM SSL do Sistema para validar os certificados de emissor e destinatário. É possível localizar a documentação detalhada sobre a validação do certificado SSL do Sistema no manual z/OS Cryptographic Services System Secure Sockets Layer Programming .

Para ativar a verificação de CRL, você especifica o local de um arquivo de configuração de CRL através do DD CRLFILE na tarefa iniciada JCL para o espaço de endereço do AMS. Um arquivo de configuração CRL de amostra que pode ser customizado é fornecido em *thlqual*.SCSQPROC(CSQ40CRL). As configurações permitidas neste arquivo são as a seguir:

Tabela 105. Variáveis de configuração de CRL do Advanced Message Security		
Variável	**Valores Válidos**	**Descrição**
crl.ldap.host[.n]	*hostname -or- hostname:port*	O ipaddr/hostname do seu servidor LDAP que hospeda os CRLs dos seus certificados de emissor. Se você não especificar um número de porta para seu servidor LDAP, o número da porta especificado pelo crl.ldap.port será usado.
crl.ldap.port	*port*	O número da porta TCP/IP do seu servidor LDAP.
crl.ldap.user	*ldap_user*	O nome de usuário LDAP a ser usado ao se conectar ao servidor LDAP.
crl.ldap.pass	*ldap_password*	A senha do LDAP associada ao crl.ldap.user.

É possível especificar vários nomes de host do servidor e portas do LDAP, conforme a seguir:

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

É possível especificar até 10 nomes de hosts. Se você não especificar um número de porta para os servidores LDAP, o número da porta especificado pela crl.ldap.port será usado. Cada servidor LDAP deve usar a mesma combinação crl.ldap.user/password para acesso.

Quando o CRLFILE DD for especificado, a configuração será carregada durante a inicialização do espaço de endereço do Advanced Message Security e a verificação de CRL será ativada. Se o CRLFILE DD não for especificado ou o arquivo de configuração do CRL estiver indisponível ou inválido, a verificação de CRL será desativada.

O AMS executa uma verificação de CRL usando os serviços certificado de validação do SSL do Sistema do IBM conforme a seguir:

Tabela 106. Verificações de CRL do Advanced Message Security		
Operação	**Qualidade de Proteção**	**Certificado(s) verificado(s)**
PUT	Privacidade	Destinatário(s)
GET	Integridade/Privacidade	Emissor

Se uma operação de mensagem falhar uma verificação de CRL Advanced Message Security executa as ações a seguir:

Tabela 107. Comportamento de falha da verificação de CRL do Advanced Message Security	
Operação	**Falha na verificação de CRL**
PUT	A mensagem não é colocada na fila de destino. Um código de conclusão MQCC_FAILED e um código de razão de MQRC_SECURITY_ERROR serão retornados para o aplicativo.

Tabela 107. Comportamento de falha da verificação de CRL do Advanced Message Security (continuação)	
Operação	**Falha na verificação de CRL**
GET	A mensagem é removida da fila de destino e movida para a fila de erros de proteção do sistema. Um código de conclusão MQCC_FAILED e um código de razão de MQRC_SECURITY_ERROR serão retornados para o aplicativo.

O AMS para o z/OS usa serviços do IBM SSL do Sistema para validar certificados, o que inclui verificações de CRL e de confiança.

O IBM MQ usa uma configuração de segurança em que a validação de certificado requer que o servidor LDAP possa ser contatado, mas não requer que uma CRL seja definida

Nota: É responsabilidade dos administradores assegurar que serviços LDAP estejam disponíveis e manter as entradas de CRL para Autoridades de certificação relevantes.

Configurando a proteção de senha do AMS para arquivos de configuração

Armazenar keystores e senhas de chaves privadas como texto sem formatação representa um risco de segurança, portanto o Advanced Message Security fornece uma ferramenta que pode misturar essas senhas usando uma chave do usuário.

Antes de começar

O proprietário do arquivo `keystore.conf` deve garantir que apenas o proprietário do arquivo tenha autorização a ler e gravar no arquivo. A proteção de senhas descrita neste tópico é apenas uma medida adicional de proteção. Além disso, é necessário executar esse procedimento em um sistema seguro.

Assegure-se de usar a variante **runamscred** correta para o tipo de cliente AMS que irá ler o arquivo de configuração. Se o cliente do AMS for um:

- Cliente Java, é necessário usar o comando Java **runamscred**, que está localizado em `<IBM MQ installation root>/java/bin`
- cliente MQI, você deve usar o comando MQI **runmqascred** que está localizado em `<IBM MQ installation root>/bin`

Procedimento

1. Edite os arquivos `keystore.conf` para incluir todas as informações necessárias, incluindo as senhas que requerem proteção.

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = no
jceks.keystore_pass = passw0rd
jceks.key_pass = passw0rd
jceks.provider = IBMJCE
```

2. Coloque a chave de criptografia para criptografar as senhas dentro de um arquivo acessível para o usuário que está protegendo o arquivo `keystore.conf`.

   Essa chave deve ser a mesma chave que será usada pelo cliente AMS posteriormente:

```
ThisIsAnExampleEncryptionKey
```

3. Execute o comando **runamscred** para proteger o arquivo `keystore.conf` fornecendo o arquivo-chave de criptografia.

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. Verifique se o arquivo `keystore.conf` foi protegido e contém senhas criptografadas.

Exemplo

O exemplo a seguir mostra como se parece um arquivo `keystore.conf` protegido:

```
jceks.keystore = c:/Documents and Settings/Alice/AliceKeystore
jceks.certificate = AliceCert
jceks.encrypted = yes
jceks.keystore_pass =
<AMS>1!62K/a4RinT+bks4RjFWx4A==!Vhi/RjIN2FH5qStUJ/0hsgKyn2IdMuhanemRRDrJq
HM=
jceks.key_pass =
<AMS>1!qmnxY++rsOUtZfDSgwcR1g==!VmWVREdVkNp1xYJstvuW64ph5vxxf7SPoqtsXxYh2
Tk=
jceks.provider = IBMJCE
```

Informações relacionadas

runamscred: proteja as palavras-chave AMS

z/OS Using certificates with AMS on z/OS

About this task

Advanced Message Security implements three levels of protection: integrity, confidentiality, and privacy.

With an integrity policy, messages are signed using the private key of the originator (the application doing the MQPUT). Integrity provides detection of message modification, but the message text itself is not encrypted.

With a confidentiality policy, the message is encrypted when it is put to the queue. The message is encrypted using a symmetric key and an algorithm specified in the relevant Advanced Message Security policy. The symmetric key itself is encrypted with the public key of each recipient (the application doing the MQGET). Public keys are associated with certificates stored in key rings.

With a privacy policy, messages are both signed and encrypted.

When a message that is protected with privacy is dequeued by a recipient application doing an MQGET, the message must be decrypted. Because it was encrypted using the recipient's public key, it must be decrypted using the recipient's private key found in a key ring.

z/OS *Use of SAF key rings with AMS on z/OS*

Advanced Message Security (AMS) makes use of z/OS SAF key ring services to define and manage the certificates needed for signing and encryption. Security products that are functionally equivalent to RACF may be used instead of RACF if they provide the same level of support.

Efficient use of key rings can reduce the administration needed to manage the certificates.

After a certificate is generated (or imported), it must be connected to a key ring to become accessible. The same certificate can be connected to more than one key ring.

Advanced Message Security uses two sets of key rings. One set consists of key rings owned by the individual user IDs that originate or receive messages. Each key ring contains the private key associated with the certificate of the owning user ID. The private key of each certificate is used to sign messages for integrity protected or privacy protected queues. It is also used to decrypt messages from privacy protected or confidentiality protected queues when receiving messages.

The other set is a single key ring owned by the AMS address space user. It contains the chain of signing CA certificates necessary to validate the certificates of the message originator and recipients.

When privacy or confidentiality protection is used, the key ring owned by the AMS address space user also contains the certificates of the message recipients. The public keys in these certificates are used to encrypt the symmetric key that was used to encrypt the message data when the message was put to the protected queue. When these messages are retrieved, the private key of relevant recipients is used to decrypt the symmetric key which is then used to decrypt the message data.

Advanced Message Security uses a key ring name of **drq.ams.keyring** when searching for certificates and private keys. This is the case for both the user and the AMS address space key rings.

For an illustration and further explanation of certificates and key ring, and their role in data protection, refer to Summary of the certificate-related operations.

The private key used for signing can have any label but must be connected as the default certificate. The private key or keys used for decryption can have any label, and must be connected to the key ring, but are no longer required to be connected as the default certificate.

Digital certificates and key rings are managed in RACF primarily by using the RACDCERT command.

For more information about certificates, labels, and the RACDCERT command, see the *z/OS: Security Server RACF Command Language Reference* and the *z/OS: Security Server RACF Security Administrator's Guide*.

z/OS *Replacing certificates*

When a certificate is renewed or replaced (for example, when the existing certificate is approaching its expiry date), it is not always possible to remove the protection from existing messages that are already on queues protected by confidentiality or privacy policies.

This can occur when the certificate was:

- Renewed with the same private key, and the reissued certificate has replaced the original certificate
- Re-keyed with a new private key and the RACDCERT ROLLOVER command has deleted the original private key

Messages will be decrypted, provided the necessary certificate is connected to the keyring of the user; it is no longer required to be connected as the default. This allows messages already on the queue, when the new certificate is connected, to be successfully decrypted.

The following example shows how a new certificate can be generated based on the existing certificate:

- A new certificate is created based on the existing certificate, with new public/private key pair.
- The new certificate is signed by the issuing authority.
- The public key of the old certificate is removed from the keyring of the AMS address space, and the public key of the new certificate is added.
- The new certificate and private key is added to the keyring of the user, in addition to the old certificate.

```
RACDCERT ID(user1) REKEY(LABEL('user1'))          -
         WITHLABEL('user1new')

RACDCERT GENREQ(LABEL('user1new')) ID(user1)      -
         DSN(output_data_set_name)

RACDCERT GENCERT(output_data_set_name) ID(user1)  -
         SIGNWITH(CERTAUTH LABEL('AMSCA'))

RACDCERT ID(user1) ALTER (LABEL('user1new'))      -
         TRUST

RACDCERT ID(WMQAMSD) REMOVE(ID(user1)             -
         LABEL('user1')                           -
         RING(drq.ams.keyring)  )

RACDCERT ID(WMQAMSD) CONNECT(ID(user1)            -
         LABEL('user1new') USAGE(SITE)            -
         RING(drq.ams.keyring)  )

RACDCERT ID(user1) CONNECT(ID(user1)              -
         LABEL('user1new') USAGE(PERSONAL)        -
         RING(drq.ams.keyring) DEFAULT )
```

For more information about certificates, labels, and the RACDCERT command, see the *z/OS: Security Server RACF Command Language Reference* and the *z/OS: Security Server RACF Security Administrator's Guide*.

z/OS *Authorizing access to the RACDCERT command for AMS on z/OS*

Authorization to use the RACDCERT command is a post-installation task that should have been completed by your z/OS system programmer. This task involves granting relevant permissions to the Advanced Message Security security administrator.

As a summary, these commands are needed to allow access to the RACF RACDCERT command:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID( admin ) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

In this example, *admin* specifies the user ID of your security administrator, or any user you want to use the RACDCERT command.

z/OS *Creating the certificates and key rings for AMS users on z/OS*

This section documents the steps required to create the certificates and key rings necessary for z/OS users of Advanced Message Security (AMS), using a RACF Certificate Authority (CA).

Resolving problems with certificates when using Advanced Message Security on z/OS

If you are having problems with certificates and missing entries in key stores you can enable a IBM Global Security Kit (GSKit) trace.

In the file referenced by the ENVARS DD in the AMS started task procedure, add:

```
GSK_TRACE_FILE=/u/... /gsktrace
GSK_TRACE=0xff
```

See Environment variables for more information.

For every access to the keystore, data is written to the trace file specified in `GSK_TRACE_FILE`.

To format the trace file use the command:

```
gsktrace inputtrace file > output_file
```

Scenario

A scenario of a sending application and a receiving application is used to explain the required steps.

In the examples that follow, `user1` is the originator of a message and `user2` is the recipient. The user ID of the Advanced Message Security address space is `WMQAMSD`.

All of the commands in the examples shown here are issued from ISPF option 6 by the administrative user ID `admin`.

z/OS *Defining a local Certificate Authority certificate for AMS on z/OS*

If you are using RACF as your CA, you must create a certificate authority certificate, if you have not already done so. The command shown here creates a certificate authority (or signer) certificate. This example creates a certificate called AMSCA to be used when creating subsequent certificates that reflect the identity of Advanced Message Security users and applications.

This command may be modified, specifically `SUBJECTSDN`, to reflect the naming structure and conventions used at your installation:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('AMSCA') O('ibm') C('us'))
KEYUSAGE(CERTSIGN) WITHLABEL('AMSCA')
```

Note: Certificates signed with this local certificate authority certificate show an issuer of CN=AMSCA,O=ibm,C=us when listed with the RACDCERT LIST command.

z/OS *Creating a digital certificate with a private key for AMS on z/OS*

A digital certificate with a private key must be generated for each Advanced Message Security user. In the example shown here, RACDCERT commands are used to generate certificates for user1 and user2, which are signed with the local CA certificate identified by the label AMSCA.

```
RACDCERT ID(user1) GENCERT SUBJECTSDN(CN('user1') O('ibm') C('us'))
WITHLABEL('user1') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user2) GENCERT SUBJECTSDN(CN('user2') O('ibm') C('us'))
WITHLABEL('user2') SIGNWITH(CERTAUTH LABEL('AMSCA'))
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

RACDCERT ID(user1) ALTER (LABEL('user1')) TRUST
RACDCERT ID(user2) ALTER (LABEL('user2')) TRUST
```

The RACDCERT ALTER command is required to add the TRUST attribute to the certificate. When a certificate is first created using this procedure, it has a different valid date range than the signing certificate. As a result, RACF marks it as NOTRUST, which means that the certificate is not to be used. Use the RACDCERT ALTER command to set the TRUST attribute.

The KEYUSAGE attributes HANDSHAKE, DATAENCRYPT and DOCSIGN must be specified for certificates used by Advanced Message Security.

Table 108. RACDCERT KEYUSAGE values and indicators

KEYUSAGE Value	**Indicators Set**
HANDSHAKE	digitalSignature and keyEncipherment
DATAENCRYPT	dataEncipherment
DOCSIGN	nonRepudiation
CERTSIGN	keyCertSign and cRLSign

z/OS *Creating the RACF key rings for AMS on z/OS*

The commands shown here create a key ring for RACF-defined user IDs user1, user2, and the Advanced Message Security address space task user WMQAMSD. The key ring name is fixed by Advanced Message Security and must be coded as shown, without quotes. The name is case-sensitive.

```
RACDCERT ID(user1) ADDRING(drq.ams.keyring)
RACDCERT ID(user2) ADDRING(drq.ams.keyring)
RACDCERT ID(WMQAMSD) ADDRING(drq.ams.keyring)
```

z/OS *Connecting the certificates to the key rings for AMS on z/OS*

Connect the user and CA certificates to the key rings:

```
RACDCERT ID(WMQAMSD) CONNECT(CERTAUTH LABEL('AMSCA')
RING(drq.ams.keyring))
RACDCERT ID(user1) CONNECT(ID(user1) LABEL('user1')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(user2) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
RACDCERT ID(WMQAMSD) CONNECT(ID(user2) LABEL('user2')
RING(drq.ams.keyring) USAGE(SITE))
```

Any certificates containing the private key or keys used for decryption must be connected to the key ring of the user, however they are no longer required to be connected as the default certificate.

The RACDCERT USAGE(SITE) attribute prevents the private key from being accessible in the key ring, while the RACDCERT USAGE(PERSONAL) attribute allows the private key to be used, if it exists. User2's certificate must be connected to the Advanced Message Security address space key ring because its public key is needed to encrypt messages as they are put to the queue. USAGE(SITE) limits exposure of user2's private key.

The CERTAUTH certificate with label AMSCA must be connected to the Advanced Message Security address space key ring because it was used to sign the certificate of user1, who is the message originator. It is used to validate user1's signing certificate.

z/OS *Key ring verification for AMS on z/OS*

The key ring should appear as shown here, after all commands have been entered:

```
RACDCERT ID(user1) LISTRING(drq.ams.keyring)
Digital ring information for user USER1:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user1                          ID(USER1)    PERSONAL YES

RACDCERT ID(user2) LISTRING(drq.ams.keyring)
Digital ring information for user USER2:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
user2                          ID(USER2)    PERSONAL YES

RACDCERT ID(WMQAMSD) LISTRING(drq.ams.keyring)
Digital ring information for user WMQAMSD:
Ring:>drq.ams.keyring<:

Certificate Label Name           Cert Owner   USAGE    DEFAULT
------------------------------ ------------ -------- -------
AMSCA                          CERTAUTH     CERTAUTH NO
user2                          ID(USER2)    SITE     NO
```

Listing the individual certificates also shows the ring association.

```
RACDCERT ID(user2) LIST(label('user2'))
Digital certificate information for user USER2:

***
Label: user2
Certificate ID: 2QfH8Pny9/LzpKKFmfFA
Status: TRUST
Start Date: 2010/05/03 22:59:53
End Date:  2011/05/04 22:59:52
Serial Number:>15<:
Issuer's Name:>OU=AMSCA.O=ibm.C=us<:
Subject's Name:>CN=user2.O=ibm.C=us<:
Key Usage: HANDSHAKE, DATAENCRYPT, DOCSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
Ring Owner: USER2
Ring:>drq.ams.keyring<:
Ring Owner: WMQAMSD
Ring:>drq.ams.keyring<:
```

To improve performance, the contents of the drq.ams.keyring associated with the AMS address space is cached for the life of the address space. Changes to that key ring do not become effective automatically. The administrator can refresh the cache by either:

- Stopping and restarting the queue manager.

- Using the z/OS MODIFY command:

```
F qmgrAMSM,REFRESH KEYRING
```

Related tasks

Operating Advanced Message Security

z/OS *Summary of the certificate-related operations for AMS on z/OS*

Figure 35 on page 679 illustrates the relationships between sending and receiving applications and relevant certificates. The scenario illustrated involves remote queuing between two z/OS queue managers using a data-protection policy of privacy. In Figure 35 on page 679, "AMS" indicates " Advanced Message Security".

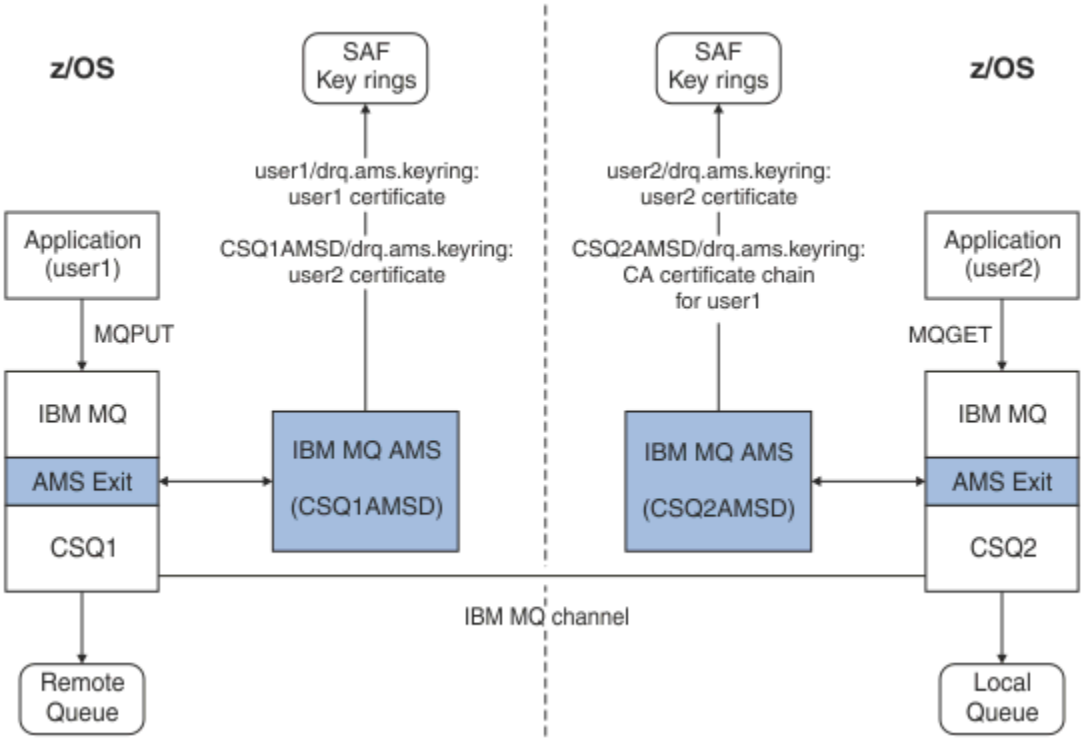


Figure 35. Application and certificate relationships

In this diagram, an application running as 'user1' puts a message to a remote queue managed by queue manager CSQ1, intended to be retrieved by an application running as 'user2' from a local queue managed by queue manager CSQ2. The diagram assumes an Advanced Message Security policy of privacy, which means the message is both signed and encrypted.

Advanced Message Security intercepts the message when a put occurs and uses user2's certificate (stored in the AMS address space user's key ring) to encrypt a symmetric key used to encrypt the message data.

Note that user2's certificate is connected to the AMS address space user key ring with option USAGE(SITE). This means the AMS address space user can access the certificate and public key, but not the private key.

On the receiving end, Advanced Message Security intercepts the get issued by user2, and uses user2's certificate to decrypt the symmetric key so that it can decrypt the message data. It then validates user1's signature using the CA certificate chain of user1's certificate stored in the AMS address space user's key ring.

Given this scenario, but with a data-protection policy of integrity, certificates for user2 would not be required.

To use Advanced Message Security to enqueue messages on IBM MQ-protected queues having a message protection policy of privacy or integrity, Advanced Message Security must have access to these data items:

- The X.509 V2 or V3 certificate and private key for the user enqueuing the message.
- The chain of certificates used to sign the digital certificates of all message signers.
- If the data protection policy is privacy, the X.509 V2 or V3 certificate of the intended recipients. The intended recipients are listed in the Advanced Message Security policy associated with the queue.

For processes and applications that run on z/OS, Advanced Message Security must have certificates in two places:

- In a SAF-managed key ring associated with the RACF identity of the sending application (the application that enqueues the protected message) or receiving application (if using privacy).

  The certificate that Advanced Message Security locates is the default certificate, and must include the private key. Advanced Message Security assumes the z/OS user identity of the sending application. That is, it acts as a surrogate, so it can access the user's private key.
- In a SAF-managed key ring associated with the AMS address space user.

  When sending messages protected with privacy, this key ring contains the public key certificates of the message recipients. When receiving messages, it contains the chain of Certificate Authority certificates needed to validate the message sender's signature.

The earlier examples shown have used RACF as the local CA. However, you may use another PKI provider (Certificate Authority) at your installation. If you intend to use another PKI product, remember that the private key and the certificate must be imported into a key ring associated with the z/OS RACF user IDs that originate IBM MQ messages protected by Advanced Message Security.

You can use the RACF RACDCERT command as the mechanism to generate certificate requests, which can be exported and sent to the PKI provider of your choice to be issued.

Here is a summary of the certificate-related steps:

1. Request the creation of a CA certificate, one in which RACF is the local CA. Omit this step if you are using another PKI provider.
2. Generate user certificates signed by the CA.
3. Create the key rings for the users and the Advanced Message Security AMS address space ID.
4. Connect the user certificate to the user key ring with the default attribute.
5. Connect the recipients certificates to the Advanced Message Security AMS address space user key ring using the usage(site) attribute (This step is necessary only for user certificates that will ultimately be the recipients of privacy-protected messages).
6. Connect the CA certificate chains for message senders to the Advanced Message Security AMS address space user key ring. (This step is necessary only for AMS tasks that will be verifying sender signatures.)

z/OS *Configuring a non-z/OS resident PKI for AMS*

Advanced Message Security for z/OS, uses X.509 V3 digital certificates in the protection-processing of messages placed on or received from IBM MQ queues. Advanced Message Security itself does not create or manage the life cycle of these certificates; that function is provided by a public key infrastructure (PKI). The examples in this publication that illustrate the use of certificates use z/OS Security Server RACF to fill certificate requests.

Whether or not a z/OS or non-z/OS resident PKI is used, AMS for z/OS uses only key rings that are managed by RACF or its equivalent. These key rings are based on Security Authorization Facility (SAF) and

are the repository used by AMS for z/OS to retrieve certificates for originators and recipients of messages placed on or received from IBM MQ queues.

For messages that are originated from z/OS, which are protected by either integrity or encryption policy, the certificate and private key of the originating user ID must be stored in an SAF-managed key ring that is associated with the z/OS user ID of the message originator.

RACF includes the capability for importing certificates and private keys into RACF-managed key rings. See the z/OS Security Server RACF publications for the details and examples of how to load certificates to RACF managed key rings.

If your installation is using one of the supported PKI products, refer to the publications that accompany the product for information on how to deploy it.

Administrando políticas de segurança do Advanced Message Security

O Advanced Message Security usa políticas de segurança para especificar a criptografia criptográfica e os algoritmos de assinatura para criptografar e autenticar mensagens que fluem através das filas.

Visão geral das políticas de segurança para AMS

As políticas de segurança do Advanced Message Security são objetos conceituais que descrevem a maneira como uma mensagem é criptograficamente criptografada e assinada.

Para obter detalhes sobre atributos da política de segurança, consulte os subtópicos a seguir:

Conceitos relacionados

"Qualidade de proteção no AMS" na página 685
A proteção de políticas do Advanced Message Security implicam uma qualidade de proteção (QOP).

"Atributos de política de segurança no AMS" na página 684
É possível usar o Advanced Message Security para selecionar um determinado algoritmo ou método para proteger os dados.

Nomes de política no AMS

O nome da política é um nome exclusivo que identifica uma política específica do Advanced Message Security e a fila para a qual ela se aplica.

O nome da política deve ser o mesmo que o nome da fila para a qual ela deve ser aplicada. Há um mapeamento um-para-um entre uma política Advanced Message Security (AMS) e uma fila.

Ao criar uma política com o mesmo nome de uma fila, você ativa a política para essa fila. As filas sem correspondência de nomes de política não são protegidas pelo AMS.

O escopo da política é relevante ao gerenciador de filas local e suas filas. Os gerenciadores de filas remotas devem ter suas próprias políticas definidas localmente para as filas que eles gerenciam.

Algoritmo de assinatura no AMS

O algoritmo de assinatura indica o algoritmo que deve ser usado ao assinar mensagens de dados.

Valores válidos são:

- MD5
- SHA-1
- família: SHA-2
  - SHA256
  - SHA384 (comprimento da chave mínimo aceitável - 768 bits)
  - SHA512 (comprimento da chave mínimo aceitável - 768 bits)

Uma política que não especifica um algoritmo de assinatura ou especifica um algoritmo de NONE, implica que as mensagens colocadas na fila associada à política não são assinadas.

Nota: A qualidade de proteção usada para as funções put e get de mensagem deve corresponder. Se houver uma qualidade de política de incompatibilidade de proteção entre a fila e a mensagem na fila, a mensagem não será aceita e será enviada à fila de manipulação de erros. Essa regra se aplica para ambas as filas locais e remotas.

Algoritmo de criptografia no AMS

O algoritmo de criptografia indica o algoritmo que deve ser usado ao criptografar mensagens de dados colocadas na fila associada à política.

Valores válidos são:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

Uma política que não especifica um algoritmo de criptografia ou especifica um algoritmo de NONE implica que mensagens colocadas na fila associada à política não serão criptografadas.

Observe que uma política que especifica um algoritmo de criptografia diferente de NONE também deve especificar pelo menos um DN de destinatário e um algoritmo de assinatura porque as mensagens criptografadas do Advanced Message Security também são assinadas.

Importante: A qualidade de proteção usada para as funções put e get de mensagem deve corresponder. Se houver uma qualidade de política de incompatibilidade de proteção entre a fila e a mensagem na fila, a mensagem não será aceita e será enviada à fila de manipulação de erros. Essa regra se aplica para ambas as filas locais e remotas.

Tolerância no AMS

O atributo tolerância indica se o Advanced Message Security pode aceitar mensagens com nenhuma política de segurança especificada.

Ao recuperar uma mensagem de uma fila com uma política para criptografar mensagens, se a mensagem não for criptografada, ela será retornada ao aplicativo de chamada. Valores válidos são:

0
Não (**padrão**).

1
Sim.

Uma política que não especifica um valor de tolerância ou especifica 0, significa que as mensagens colocadas na fila associada à política devem corresponder às regras de política.

A tolerância é opcional e existe para facilitar o roll-out de configuração, no qual as políticas foram aplicadas a filas, mas essas filas já contêm mensagens que não possuem uma política de segurança especificada.

Nomes distintos do remetente no AMS

Os nomes distintos (DNs) do emissor identificam usuários que estão autorizados a colocar mensagens em uma fila. Um emissor usa seu certificado para assinar uma mensagem antes de colocar a mensagem em uma fila.

O Advanced Message Security (AMS) não verifica se uma mensagem foi colocada em uma fila protegida por dados por um usuário válido até que a mensagem seja recuperada. Neste momento, se a política estipular um ou mais emissores válidos e o usuário que colocou a mensagem na fila não estiver na lista de emissores válidos, o AMS retornará um erro para o aplicativo de recebimento e colocará a mensagem na fila de erros AMS

Uma política pode ter zero ou mais DNs do emissor especificado. Se nenhum DNs do emissor for especificado para a política, qualquer emissor poderá colocar mensagens protegidas por dados na fila, desde que o certificado do emissor seja confiável. Um certificado do remetente é confiável incluindo o certificado público em um keystore disponível para o aplicativo de recebimento

Nomes distintos do emissor têm o formato a seguir:

```
CN=Common Name,O=Organization,C=Country
```

Importante:

- Todos os nomes de Componentes DN devem estar em maiúsculas. Todos os identificadores de nome do componente no DN devem ser especificados na ordem mostrada na tabela a seguir:

Nome do Componente	Value
CN	O nome comum para o objeto deste DN, como um nome completo ou a finalidade de um dispositivo.
OU	A unidade dentro da organização com o qual o objeto do DN é afiliado, como uma divisão corporativa ou um nome do produto.
O	A organização com a qual o objeto do DN está afiliado, como uma corporação.
L	A localidade (cidade ou municipalidade) onde o objeto do DN está localizado.
Estado	O nome do estado ou da província onde o objeto do DN está localizado.
C	O país onde o objeto do nome distinto (DN) está localizado.

- Se um ou mais DNs de remetentes forem especificados para a política, somente aqueles usuários poderão colocar mensagens na fila associada com a política.
- Os DNs de remetentes, quando especificados, devem corresponder exatamente ao DN contido no certificado digital associado com o usuário que coloca a mensagem.
- O AMS suporta DNs somente com valores do conjunto de caracteres Latin-1. Para criar DNs com caracteres do conjunto, deve-se primeiro criar um certificado com um DN que é criado na codificação UTF-8 usando AIX and Linux com a codificação UTF-8 ativada. Em seguida, deve-se criar uma política por meio de uma plataforma Linux ou AIX com a codificação UTF-8 ativada ou usar o plug-in AMS no IBM MQ.
- O método usado pelo AMS, para converter o nome do emissor de formato x.509 para DN, sempre usa ST= para o valor State ou Province.
- Os seguintes caracteres especiais precisam de caracteres de escape:

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- Se o Nome distinto contiver espaços em branco integrados, será necessário incluir o DN entre aspas duplas.

Conceitos relacionados

“Nomes distintos do destinatário no AMS” na página 684

Os nomes distintos (DN) do destinatário identificam usuários que estão autorizados a recuperar mensagens em uma fila.

Nomes distintos do destinatário no AMS

Os nomes distintos (DN) do destinatário identificam usuários que estão autorizados a recuperar mensagens em uma fila.

Uma política pode ter zero ou mais DNs do destinatário especificado. Os nomes distintos do destinatário têm o formato a seguir:

```
CN=Common Name,O=Organization,C=Country
```

Importante:

- Todos os nomes de Componentes DN devem estar em maiúsculas. Todos os identificadores de nome do componente no DN devem ser especificados na ordem mostrada na tabela a seguir:

Nome do Componente	Value
CN	O nome comum para o objeto deste DN, como um nome completo ou a finalidade de um dispositivo.
OU	A unidade dentro da organização com o qual o objeto do DN é afiliado, como uma divisão corporativa ou um nome do produto.
O	A organização com a qual o objeto do DN está afiliado, como uma corporação.
L	A localidade (cidade ou municipalidade) onde o objeto do DN está localizado.
Estado	O nome do estado ou da província onde o objeto do DN está localizado.
C	O país onde o objeto do nome distinto (DN) está localizado.

- Se nenhum DN de destinatário for especificado para a política, qualquer usuário poderá receber mensagens da fila associada com a política.
- Se um ou mais DNs do destinatário forem especificados para a política, apenas esses usuários poderão obter mensagens da fila associada à política.
- Os DNs de destinatários, quando especificados, devem corresponder exatamente ao DN contido no certificado digital associado com o usuário que recebe a mensagem.
- O Advanced Message Security suporta DNs somente com valores do conjunto de caracteres Latin-1. Para criar DNs com caracteres do conjunto, deve-se primeiro criar um certificado com um DN que é criado na codificação UTF-8 usando AIX ou Linux com a codificação UTF-8 ativada. Em seguida, deve-se criar uma política por meio de uma plataforma Linux ou AIX com a codificação UTF-8 ativada ou usar o plug-in Advanced Message Security no IBM MQ.

Conceitos relacionados

“Nomes distintos do remetente no AMS” na página 682
Os nomes distintos (DNs) do emissor identificam usuários que estão autorizados a colocar mensagens em uma fila. Um emissor usa seu certificado para assinar uma mensagem antes de colocar a mensagem em uma fila.

Atributos de política de segurança no AMS

É possível usar o Advanced Message Security para selecionar um determinado algoritmo ou método para proteger os dados.

Uma política de segurança é um objeto conceitual que descreve a maneira como uma mensagem é criptograficamente criptografada e assinada.

Tabela 109. Atributos de política de segurança no AMS

Atributos	**Descrição**
Nome da política	Nome exclusivo da política para um gerenciador de filas.
Algoritmo de Assinatura	O algoritmo criptográfico que é usado para assinar mensagens antes do envio.
Algoritmo de criptografia	O algoritmo criptográfico que é usado para criptografar mensagens antes do envio.
Lista de destinatários	Lista de nomes distintos (DNs) de certificado dos destinatários em potencial de uma mensagem.
Lista de verificação do DN de assinatura	Lista de DNs de assinatura para serem validados durante a recuperação da mensagem.

No Advanced Message Security, as mensagens são criptografadas com uma chave simétrica e a chave simétrica é criptografada com as chaves públicas dos destinatários. Chaves públicas são criptografadas com o algoritmo RSA, com chaves de um tamanho efetivo de até 2048 bits. A criptografia de chave assimétrica real depende do comprimento da chave do certificado.

Os algoritmos de chave simétrica suportados são os seguintes:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

O Advanced Message Security também suporta as funções hash de criptografia a seguir:

- Deprecated MD5
- Deprecated SHA-1
- família: SHA-2
  - SHA256
  - SHA384 (comprimento da chave mínimo aceitável - 768 bits)
  - SHA512 (comprimento da chave mínimo aceitável - 768 bits)

Nota: A qualidade de proteção usada para as funções put e get de mensagem deve corresponder. Se houver uma qualidade de política de incompatibilidade de proteção entre a fila e a mensagem na fila, a mensagem não será aceita e será enviada à fila de manipulação de erros. Essa regra se aplica para ambas as filas locais e remotas.

Qualidade de proteção no AMS

A proteção de políticas do Advanced Message Security implicam uma qualidade de proteção (QOP).

Os três níveis de qualidade de proteção no Advanced Message Security são suplementados por um quarto nível no IBM MQ 9.0 e mais recente e todos eles dependem de algoritmos criptográficos que são usados para assinar e criptografar a mensagem:

- Privacidade - as mensagens colocadas na fila devem ser assinadas e criptografadas.
- Integridade - as mensagens colocadas na fila devem ser assinadas pelo emissor.

- Confidencialidade - as mensagens colocadas na fila devem ser criptografadas. Para obter informações adicionais, consulte “Qualidades de proteção disponíveis com AMS” na página 609
- Nenhum - Nenhuma proteção de dados é aplicável.

Uma política que determina que as mensagens devem ser assinadas quando colocadas em uma fila tem um QOP de INTEGRITY. Um QOP de INTEGRITY significa que uma política estipula um algoritmo de assinatura, mas não estipula um algoritmo de criptografia. As mensagens protegidas por integridade são também referidas como "ASSINADAS".

Uma política que determina que as mensagens devem ser assinadas e criptografadas quando colocadas em uma fila tem um QOP de PRIVACY. Um QOP de PRIVACIDADE significa que uma política estipula um algoritmo de assinatura e um algoritmo de criptografia. As mensagens protegidas por privacidade são também referidas como "SELADAS".

Uma política que estipula que as mensagens deverão ser criptografadas quando colocadas em uma fila tem uma QOP de CONFIDENCIALIDADE. Uma QOP de CONFIDENCIALIDADE significa que uma política estipula um algoritmo de criptografia.

Uma política que não estipula um algoritmo de assinatura ou um algoritmo de criptografia tem um QOP de NONE. O Advanced Message Security não fornece proteção de dados para as filas que possuem uma política com um QOP de NONE.

Gerenciando políticas de segurança no AMS

Uma política de segurança é um objeto conceitual que descreve a maneira como uma mensagem é criptograficamente criptografada e assinada.

O local do qual todas as tarefas administrativas relacionadas às políticas de segurança são executadas depende de qual plataforma você está usando.

- ALW No AIX, Linux, and Windows, você usa os comandos DELETE POLICY, DISPLAY POLICY e SET POLICY (ou PCF equivalente) para gerenciar suas políticas de segurança.
  - Linux AIX No AIX and Linux, as tarefas administrativas podem ser executadas em *MQ_INSTALLATION_PATH*/bin.
  - Windows Nas plataformas Windows, as tarefas administrativas podem ser executadas em qualquer local porque a variável de ambiente PATH é atualizada na instalação.
- IBM i No IBM i, os comandos DSPMQMSPL, SETMQMSPL e WRKMQMSPL serão instalados na biblioteca do sistema QSYS para o idioma principal do sistema quando o IBM MQ for instalado.

  As versões de idioma nacional adicionais serão instaladas nas bibliotecas QSYS29xx de acordo com o carregamento do recurso de idioma. Por exemplo, uma máquina com o inglês americano como o idioma principal e o coreano como o idioma secundário possui os comandos em inglês americano instalado em QSYS e o carregamento de idioma secundário em coreano no QSYS2962, já que 2962 é o carregamento de idioma para coreano.
- z/OS No z/OS, os comandos administrativos são executados usando o utilitário de política de segurança da mensagem (CSQ0UTIL). Quando as políticas forem criadas, modificadas ou excluídas no z/OS, as mudanças não serão reconhecidas pelo Advanced Message Security até que o gerenciador de filas seja parado e reiniciado ou o comando MODIFY do z/OS seja usado para atualizar a configuração de política do Advanced Message Security. Por exemplo:

```
F <qmgr ssid>AMSM,REFRESH POLICY
```

Tarefas relacionadas

“Criando políticas de segurança no AMS” na página 687
As políticas de segurança definem a maneira na qual uma mensagem será protegida quando a mensagem for colocada ou como uma mensagem deve ter sido protegida quando uma mensagem é recebida.

“Alterando Políticas de Segurança no AMS” na página 688
É possível usar o Advanced Message Security para alterar detalhes de políticas de segurança que você já definiu.

“Exibindo e fazendo dump de políticas de segurança no AMS” na página 688
Use o comando **dspmqspl** para exibir uma lista de todas as políticas de segurança ou detalhes de uma política denominada dependendo dos parâmetros da linha de comandos que você fornecer.

“Removendo políticas de segurança no AMS” na página 690
Para remover políticas de segurança no Advanced Message Security, deve-se usar o comando `setmqspl`.

Operando Advanced Message Security

Referências relacionadas

O utilitário de política de segurança da mensagem (CSQ0UTIL)

Criando políticas de segurança no AMS

As políticas de segurança definem a maneira na qual uma mensagem será protegida quando a mensagem for colocada ou como uma mensagem deve ter sido protegida quando uma mensagem é recebida.

Antes de começar

Há algumas condições de entrada que devem ser atendidas ao criar políticas de segurança:

- O gerenciador de filas deve estar em execução.
- O nome de uma política de segurança deve seguir Regras para nomenclatura de objetos IBM MQ.
- Deve-se ter a autoridade necessária para se conectar ao gerenciador de filas e criar uma política de segurança:
  - z/OS No z/OS, conceda as autoridades documentadas em O utilitário de política de segurança da mensagem (CSQ0UTIL).
  - Multi Em Multiplataformas, deve-se conceder as autoridades `+connect`, `+inq` e `+chg` necessárias usando o comando **setmqaut**

  Para obter mais informações sobre como configurar a segurança, consulte “Configurar a segurança” na página 137.
- z/OS No z/OS, assegure-se de que os objetos do sistema necessários foram definidos de acordo com as definições em CSQ4INSM.

Exemplo

A seguir há um exemplo da criação de uma política no gerenciador de filas QMGR. A política especifica que as mensagens sejam assinadas usando o algoritmo SHA256 e criptografadas usando o algoritmo AES256 para certificados com DN: CN=joe,O=IBM,C=US and DN: CN=jane,O=IBM,C=US. Essa política está conectada ao `MY.QUEUE`:

```
setmqspl -m QMGR -p MY.QUEUE -s SHA256 -e AES256 -r CN=joe,O=IBM,C=US -r CN=jane,O=IBM,C=US
```

Aqui está um exemplo de criação de política no gerenciador de filas QMGR. A política especifica que as mensagens sejam criptografadas usando o algoritmo 3DES para certificados com DNs: CN=john,O=IBM,C=US and CN=jeff,O=IBM,C=US e assinadas com o algoritmo SHA256 para o certificado com DN: CN=phil,O=IBM,C=US

```
setmqspl -m QMGR -p MY.OTHER.QUEUE -s SHA256 -e 3DES -r CN=john,O=IBM,C=US -r
CN=jeff,O=IBM,C=US -a CN=phil,O=IBM,C=US
```

Nota:

- A qualidade da proteção sendo usada para a mensagem de colocação e obtenção deve corresponder. Se a qualidade da política de proteção que é definida para a mensagem for mais fraca do que a definida

para uma fila, a mensagem será enviada para a fila de manipulação de erros. Esta política é válida para as filas local e remota.

Referências relacionadas

Lista completa dos atributos do comando setmqspl

Alterando Políticas de Segurança no AMS

É possível usar o Advanced Message Security para alterar detalhes de políticas de segurança que você já definiu.

Antes de começar

- O gerenciador de filas no qual deseja operar deve estar em execução.
- Deve-se ter a autoridade necessária para se conectar ao gerenciador de filas e criar uma política de segurança.
  - z/OS No z/OS, conceda as autoridades documentadas em O utilitário de política de segurança da mensagem (CSQ0UTIL).
  - Multi Em Multiplataformas, deve-se conceder as autoridades `+connect`, `+inq` e `+chg` necessárias usando o comando **setmqaut**

  Para obter mais informações sobre como configurar a segurança, consulte “Configurar a segurança” na página 137.

Sobre esta tarefa

Para mudar as políticas de segurança, aplique o comando `setmqspl` a uma política já existente fornecendo novos atributos.

Exemplo

Aqui está um exemplo da criação de uma política denominada MYQUEUE em um gerenciador de filas denominado QMGR, especificando que as mensagens devem ser criptografadas usando o algoritmo 3DES para autores (**-a**) que possuem certificados com Nome Distinto (DN) de CN=alice, O=IBM, C=US e assinado com o algoritmo SHA256 para destinatários (**-r**) que possuem certificados com DN de CN=jeff, O=IBM, C = US.

```
setmqspl -m QMGR -p MYQUEUE -e 3DES -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

Para alterar essa política, emita o comando `setmqspl` com todos os atributos do exemplo, mudando somente os valores que você deseja modificar. Neste exemplo, a política criada anteriormente é conectada a uma nova fila e seu algoritmo de criptografia é mudado para AES256:

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

Referências relacionadas

setmqspl (configurar política de segurança)

Exibindo e fazendo dump de políticas de segurança no AMS

Use o comando **dspmqspl** para exibir uma lista de todas as políticas de segurança ou detalhes de uma política denominada dependendo dos parâmetros da linha de comandos que você fornecer.

Antes de começar

- Para exibir detalhes de políticas de segurança, o gerenciador de filas deve existir e estar em execução.
- Deve-se ter a autoridade necessária para se conectar ao gerenciador de filas e criar uma política de segurança.

– z/OS No z/OS, conceda as autoridades documentadas em O utilitário de política de segurança da mensagem (CSQ0UTIL).

– Multi Em Multiplataformas, deve-se conceder as autoridades `+connect`, `+inq` e `+chg` necessárias usando o comando **setmqaut**

Para obter mais informações sobre como configurar a segurança, consulte “Configurar a segurança” na página 137.

Sobre esta tarefa

Aqui está a lista de sinalizadores de comando **dspmqspl**:

*Tabela 110. Sinalizadores de comando **dspmqspl**.*

Sinalizador de comando	Explicação
-m	Nome do gerenciador de filas (obrigatório).
-p	Nome da política.
-export	Incluir este sinalizador gera uma saída que pode ser facilmente aplicada a um gerenciador de filas diferente.

Exemplo

O exemplo a seguir mostra como criar duas políticas de segurança para `venus.queue.manager`:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s sha256 -a "CN=signer1,O=IBM,C=US" -e NONE
setmqspl -m venus.queue.manager -p AMS_POL_06_THREE -s sha256 -a "CN=another signer,O=IBM,C=US"
-e NONE
```

Este exemplo mostra um comando que exibe detalhes de todas as políticas definidas para `venus.queue.manager` e a saída que ele produz:

```
dspmqspl -m venus.queue.manager

Policy Details:
Policy name: AMS_POL_04_ONE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=signer1,O=IBM,C=US
Recipient DNs: -
Toleration: 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

Este exemplo mostra um comando que exibe detalhes de uma política de segurança selecionada definida para `venus.queue.manager` e a saída que ela produz:

```
dspmqspl -m venus.queue.manager -p AMS_POL_06_THREE

Policy Details:
Policy name: AMS_POL_06_THREE
Quality of protection: INTEGRITY
Signature algorithm: SHA256
```

```
Encryption algorithm: NONE
Signer DNs:
  CN=another signer,O=IBM,C=US
Recipient DNs: -
Toleration: 0
```

No próximo exemplo, em primeiro lugar, criamos uma política de segurança e, em seguida, exportamos a política usando o sinalizador **-export**:

```
setmqspl -m venus.queue.manager -p AMS_POL_04_ONE -s SHA256 -a "CN=signer1,O=IBM,C=US" -e NONE

dspmqspl -m venus.queue.manager -export
```

z/OS No z/OS, as informações de política exportadas são gravadas pelo CSQ0UTIL para o EXPORT DD.

Multi Em Multiplataformas, redirecione a saída para um arquivo, por exemplo:

```
dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

Para importar uma política de segurança:

- Linux AIX No AIX and Linux:
  1. Efetue logon como um usuário que pertence ao grupo de administração mqm IBM MQ.
  2. Emita `. policies.sh`..
- Windows No Windows, execute `policies.bat`.
- z/OS No z/OS use o usuário CSQ0UTIL, especificando SYSIN para o conjunto de dados que contém as informações da política exportada.

Referências relacionadas

Lista completa dos atributos do comando dspmqspl

Removendo políticas de segurança no AMS

Para remover políticas de segurança no Advanced Message Security, deve-se usar o comando `setmqspl`.

Antes de começar

Há algumas condições de entrada que devem ser atendidas ao gerenciar políticas de segurança:

- O gerenciador de filas deve estar em execução.
- Deve-se ter a autoridade necessária para se conectar ao gerenciador de filas e criar uma política de segurança.
  - z/OS No z/OS, conceda as autoridades documentadas em O utilitário de política de segurança da mensagem (CSQ0UTIL).
  - Multi Em Multiplataformas, deve-se conceder as autoridades `+connect`, `+inq` e `+chg` necessárias usando o comando **setmqaut**

  Para obter mais informações sobre como configurar a segurança, consulte “Configurar a segurança” na página 137.

Sobre esta tarefa

Use o comando **setmqspl** com a opção **-remove**.

Exemplo

Aqui está um exemplo de remoção de uma política:

```
setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

Referências relacionadas

Lista completa dos atributos do comando setmqspl

Proteção da fila do sistema no AMS

As filas do sistema ativam a comunicação entre IBM MQ e seus aplicativos auxiliares. Sempre que um gerenciador de filas é criado, uma fila do sistema também é criada para armazenar mensagens internas e dados das mensagens do IBM MQ. É possível proteger filas do sistema com o Advanced Message Security para que somente usuários autorizados possam acessá-las ou decriptografá-las.

A proteção de fila do sistema segue o mesmo padrão que a proteção das filas regulares. Consulte o “Criando políticas de segurança no AMS” na página 687.

Windows Para usar a proteção de fila do sistema no Windows, copie o arquivo `keystore.conf` para o diretório a seguir:

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS No z/OS, para fornecer proteção para o `SYSTEM.ADMIN.COMMAND.QUEUE`, o servidor de comandos deve ter acesso ao `keystore` e ao `keystore.conf`, que contêm chaves e uma configuração para que o servidor de comandos possa acessar chaves e certificados Todas as mudanças feitas na política de segurança de `SYSTEM.ADMIN.COMMAND.QUEUE` requerem o reinício do servidor de comandos.

Todas as mensagens que são enviadas e recebidas da fila de comandos são assinadas ou assinadas e criptografadas dependendo das configurações de política. Se um administrador define os signatários autorizados, as mensagens de comando que não passam pela verificação do Nome distinto (DN) do assinante não são executadas pelo servidor de comandos e não são roteadas para a fila de manipulação de erros do Advanced Message Security. Mensagens que são enviadas como respostas para as filas dinâmicas temporárias do IBM MQ Explorer não são protegidas pelo AMS.

As políticas de segurança não têm efeito sobre as filas SYSTEM a seguir:

- SYSTEM.ADMIN.ACCOUNTING.QUEUE
- SYSTEM.ADMIN.ACTIVITY.QUEUE
- SYSTEM.ADMIN.CHANNEL.EVENT
- SYSTEM.ADMIN.COMMAND.EVENT
- z/OS SYSTEM.ADMIN.COMMAND.QUEUE
- SYSTEM.ADMIN.CONFIG.EVENT
- SYSTEM.ADMIN.LOGGER.EVENT
- SYSTEM.ADMIN.PERFM.EVENT
- SYSTEM.ADMIN.PUBSUB.EVENT
- SYSTEM.ADMIN.QMGR.EVENT
- SYSTEM.ADMIN.STATISTICS.QUEUE
- SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- SYSTEM.AUTH.DATA.QUEUE
- SYSTEM.BROKER.ADMIN.STREAM
- z/OS SYSTEM.BROKER.CLIENTS.DATA
- SYSTEM.BROKER.CONTROL.QUEUE
- SYSTEM.BROKER.DEFAULT.STREAM

- SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS
- z/OS SYSTEM.BROKER.SUBSCRIPTIONS.DATA
- SYSTEM.CHANNEL.INITQ
- SYSTEM.CHANNEL.SYNCQ
- z/OS SYSTEM.CHLAUTH.DATA.QUEUE
- SYSTEM.CICS.INITIATION.QUEUE
- SYSTEM.CLUSTER.COMMAND.QUEUE
- SYSTEM.CLUSTER.HISTORY.QUEUE
- SYSTEM.CLUSTER.REPOSITORY.QUEUE
- SYSTEM.CLUSTER.TRANSMIT.QUEUE
- z/OS SYSTEM.COMMAND.INPUT
- z/OS SYSTEM.DDELAY.LOCAL.QUEUE
- SYSTEM.DEAD.LETTER.QUEUE
- SYSTEM.DURABLE.SUBSCRIBER.QUEUE
- SYSTEM.HIERARCHY.STATE
- SYSTEM.INTER.QMGR.CONTROL
- SYSTEM.INTER.QMGR.FANREQ
- SYSTEM.INTER.QMGR.PUBS
- SYSTEM.INTERNAL.REPLY.QUEUE
- z/OS SYSTEM.JMS.PS.STATUS.QUEUE
- z/OS SYSTEM.JMS.REPORT.QUEUE
- SYSTEM.PENDING.DATA.QUEUE
- SYSTEM.PROTECTION.ERROR.QUEUE
- SYSTEM.PROTECTION.POLICY.QUEUE
- z/OS SYSTEM.QSG.CHANNEL.SYNCQ
- z/OS SYSTEM.QSG.TRANSMIT.QUEUE
- z/OS SYSTEM.QSG.UR.RESOLUTION.QUEUE
- SYSTEM.RETAINED.PUB.QUEUE
- z/OS SYSTEM.RETAINED.PUB.QUEUE
- SYSTEM.SELECTION.EVALUATION.QUEUE
- SYSTEM.SELECTION.VALIDATION.QUEUE

Multi Filas de fluxo e AMS

É possível transmitir mensagens protegidas do Advanced Message Security (AMS) duplicadas.

Se uma fila tiver uma política do AMS definida que faz com que as mensagens colocadas nessa fila sejam assinadas e/ou criptografadas, também será possível configurar o atributo **STREAMQ** da fila para colocar uma cópia de cada mensagem protegida em uma segunda fila. A mensagem transmitida duplicada é assinada e/ou criptografada usando a mesma política que foi configurada para a fila original.

No exemplo a seguir, você configurará duas filas, QUEUE1 e QUEUE2. QUEUE1 tem seu atributo **STREAMQ** configurado para colocar mensagens transmitidas para QUEUE2:

```
DEFINE QLOCAL(QUEUE2)
```

```
DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

As mensagens protegidas do AMS estão sendo colocadas em QUEUE1 por um usuário com o certificado `CN=bob,O=IBM,C=GB`.

Um aplicativo com certificado `CN=alice,O=IBM,C=GB` vai consumir as mensagens de QUEUE1. Um aplicativo separado com certificado `CN=fred,O=IBM,C=GB` vai consumir as mensagens de QUEUE2.

QUEUE1 tem a política de privacidade do AMS a seguir aplicada a ele:

```
SET POLICY(QUEUE1) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=alice,O=IBM,C=GB') RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

Se um algoritmo de criptografia tiver sido configurado na política para QUEUE1, os destinatários listados na política deverão incluir tanto os destinatários das mensagens originais de QUEUE1 quanto os destinatários que vão consumir mensagens duplicadas de QUEUE2.

Quando o aplicativo tenta consumir mensagens de QUEUE2, ele executa verificações de integridade, e/ou decriptografa a mensagem com base na política que foi configurada em QUEUE2. Para um aplicativo que deseja consumir mensagens transmitidas de QUEUE2, deve-se configurar uma política adequada em QUEUE2 que permita que as mensagens sejam verificadas quanto à integridade e decriptografadas corretamente.

Em particular, o algoritmo de assinatura, o assinante e o algoritmo de criptografia devem ser os mesmos da política aplicada ao QUEUE1. Os destinatários da política para QUEUE2 devem incluir a identidade do destinatário que consome a mensagem de QUEUE2.

Nota: Não é necessário que a política aplicada ao QUEUE2 liste todos os destinatários nomeados no conjunto de políticas em QUEUE1.

Por exemplo, a política a seguir poderia ser configurada em QUEUE2 para permitir que um aplicativo com o nome distinto do certificado `CN=fred,O=IBM,C=GB` leia mensagens protegidas pelo AMS por meio dele:

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

Conceitos relacionados

Filas de fluxo

Concedendo permissões do OAM no AMS

As permissões de arquivo autorizam todos os usuários a executar os comandos `setmqspl` e `dspmqspl`. No entanto, o Advanced Message Security depende do gerenciador de autoridade de objeto (OAM) e cada tentativa de executar estes comandos por um usuário que não pertence ao grupo mqm, que é o grupo de administração do IBM MQ ou não tem permissões para ler as configurações de política de segurança que são concedidas, resulta em um erro.

Procedimento

Para conceder as permissões necessárias para um usuário, execute:

```
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
```

Nota: É necessário somente configurar estas autoridades OAM se você pretende conectar clientes para o gerenciador de filas usando o Advanced Message Security 7.0.1.

Atenção: A autoridade de navegação para SYSTEM.PROTECTION.POLICY.QUEUE não é obrigatório em todas as situações. O IBM MQ otimiza o desempenho armazenando em cache políticas para que você não tenha que procurar registros para obter detalhes da política no SYSTEM.PROTECTION.POLICY.QUEUE em todos os casos..

O IBM MQ não armazena em cache todas as políticas disponíveis. Se houver um grande número de políticas, o IBM MQ armazenará em cache um número limitado de políticas. Portanto, se o gerenciador de filas tiver um número baixo de políticas definidas, não haverá a necessidade de fornecer a opção de navegação para o SYSTEM.PROTECTION.POLICY.QUEUE.

No entanto, é necessário dar autoridade de navegação a essa fila nos casos em que há um grande número de políticas definidas ou quando você está usando clientes antigos. O SYSTEM.PROTECTION.ERROR.QUEUE é usado para colocar mensagens de erro geradas pelo código AMS. A autoridade put para essa fila é verificada apenas quando você tenta colocar uma mensagem de erro na fila. Sua autoridade put em relação à fila não é verificada quando você tenta colocar ou obter mensagens de uma fila protegida por AMS.

Concedendo permissões de segurança no AMS

Ao usar a segurança de recurso do comando, deve-se configurar as permissões para permitir que o Advanced Message Security funcione. Este tópico usa comandos do RACF nos exemplos. Se a sua empresa usa um gerenciador de segurança externo (ESM) diferente, deve-se usar os comandos equivalentes para esse ESM.

Há três aspectos para a concessão de permissões de segurança:

- "O espaço de endereço AMSM" na página 694
- "CSQ0UTIL" na página 694
- "Usando filas que possuem uma política do Advanced Message Security definida" na página 695

Notas: Os exemplos de comando usam as variáveis a seguir.

1. *QMgrName* - o nome do gerenciador de filas.

   z/OS No z/OS, esse valor também pode ser o nome de um grupo de filas compartilhadas.
2. *username* - este pode ser um nome de grupo.
3. Os exemplos mostram a classe MQQUEUE. z/OS isso também pode ser MXQUEUE, GMQQUEUE ou GMXQUEUE. Consulte "Profiles for queue security" na página 207 para obter informações adicionais.

Além disso, se o perfil já existir, o comando RDEFINE não será necessário.

O espaço de endereço AMSM

É necessário emitir alguma segurança do IBM MQ para o nome do usuário no qual o espaço de endereço do Advanced Message Security é executado.

- Para conexão em lotes para o gerenciador de filas, emita

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
           PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- Para obter acesso à SYSTEM.PROTECTION.POLICY.QUEUE, emita:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

CSQ0UTIL

O utilitário que permite que os usuários executem os comandos **setmqspl** e **dspmqspl** requer as permissões a seguir, no qual o nome do usuário é o ID do usuário do cargo:

- Para conexão em lotes para o gerenciador de filas, emita:

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
           PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- Para acesso ao SYSTEM.PROTECTION.POLICY.QUEUE, necessário para o comando **setmqpol**, emita:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
              PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- Para acesso ao SYSTEM.PROTECTION.POLICY.QUEUE, necessário para o comando **dspmqpol**, emita:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
                PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

Usando filas que possuem uma política do Advanced Message Security definida

Quando um aplicativo realiza qualquer trabalho com filas que têm uma política definida nelas, o aplicativo requer permissões adicionais para permitir que o Advanced Message Security proteja as mensagens.

O aplicativo requer:

- Acesso de leitura ao SYSTEM.PROTECTION.POLICY.QUEUE. Faça isso emitindo:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- Acesso de colocação ao SYSTEM.PROTECTION.ERROR.QUEUE. Faça isso emitindo:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
            PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

IBM i Configurando certificados e o arquivo de configuração do keystore para o AMS no IBM i

Sua primeira tarefa ao configurar a proteção do Advanced Message Security é criar um certificado e associá-lo ao seu ambiente. A associação é configurada através de um arquivo mantido no sistema de arquivos integrado (IFS).

Procedimento

1. Para criar um certificado autoassinado usando o conjunto de ferramentas OpenSSL enviado com o IBM i, emita o seguinte comando a partir de QShell:

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
   $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
   ```

   O comando solicita vários atributos de nome distinto para um novo certificado autoassinado, incluindo:

   - Nome comum (CN=)
   - Organização (O=)
   - País (C=)

   Isso cria uma chave privada não criptografada e um certificado correspondente, ambos em PEM (Privacy Enhanced Mail) formato.

   Para simplificar, simplesmente insira os valores para nome comum, organização e país. Esses atributos e valores são importantes ao criar uma política.

   Prompts e atributos adicionais podem ser customizados especificando um arquivo de configuração openssl customizado na linha de comandos com o parâmetro **-config**. Consulte a documentação do OpenSSL para obter mais detalhes sobre a sintaxe do arquivo de configuração.

Por exemplo, o comando a seguir inclui extensões adicionais do certificado X.509 v3:

```
/QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048
-keyout $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365 -config myconfig.cnf
```

em que myconfig.cnf é um arquivo de fluxo ASCII que contém o seguinte:

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = myextensions

[req_distinguished_name]
countryName = Country Name (2 letter code)
countryName_default = GB
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = Hants
localityName = Locality Name (eg, city)
localityName_default = Hursley
organizationName = Organization Name (eg, company)
organizationName_default = IBM United Kingdom
organizationalUnitName = Organizational Unit Name (eg, department)
organizationalUnitName_default = IBM MQ Development
commonName = Common Name (eg, Your Name)

[myextensions]
keyUsage = digitalSignature,nonRepudiation,dataEncipherment,keyEncipherment
extendedKeyUsage = emailProtection
```

2. O AMS requer que o certificado e a chave privada sejam mantidos no mesmo arquivo. Emita o comando a seguir para fazer isso:

```
cat $HOME/mycert.pem >> $HOME/private.pem
```

O arquivo `private.pem` em $HOME agora contém uma chave privada e certificado correspondentes, enquanto o arquivo `mycert.pem` contém todos os certificados públicos para o qual é possível criptografar mensagens e validar assinaturas.

Os dois arquivos precisam ser associados ao seu ambiente criando um arquivo de configuração de keystore, `keystore.conf`, em sua localização padrão.

Por padrão, o AMS procura a configuração de keystore em um subdiretório `.mqs` do seu diretório inicial.

3. No QShell, crie o arquivo `keystore.conf`:

```
mkdir -p $HOME/.mqs
echo "pem.private = $HOME/private.pem" > $HOME/.mqs/keystore.conf
echo "pem.public = $HOME/mycert.pem" >> $HOME/.mqs/keystore.conf
echo "pem.password = unused" >> $HOME/.mqs/keystore.conf
```

IBM i *Criando uma política para o AMS no IBM i*

Antes de criar uma política, você precisa criar uma fila para manter as mensagens protegidas.

Procedimento

1. Em um prompt de linha de comandos insira;

```
CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
```

em que mqmname é o nome do gerenciador de filas.

Use o comando DSPMQM para verificar se o gerenciador de filas é capaz de usar políticas de segurança. Assegure que **Security Policy Capability** mostre **YES*.

A política mais simples que é possível definir é uma política de integridade, que é obtida criando uma política com um algoritmo de assinatura digital, mas nenhum algoritmo de criptografia.

As mensagens são assinadas, mas não criptografadas. Se as mensagens devem ser criptografadas, deve-se especificar um algoritmo de criptografia e um ou mais destinatários de mensagem desejados.

Um certificado no keystore público para um destinatário da mensagem desejada é identificada por meio de um nome distinto.

2. Exiba os nomes distintos dos certificados no keystore público, `mycert.pem` em `$HOME`, usando o comando a seguir no QShell:

```
/QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
```

   É necessário inserir o nome distinto como um destinatário-alvo e o nome da política deve corresponder ao nome da fila a ser protegido.

3. Em um prompt de comandos da CL insira, por exemplo:

```
SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
O=.., C=..')
```

   em que `mqmname` é o nome do gerenciador de filas.

   Depois de criar a política, quaisquer mensagens que sejam colocadas, procuradas ou removidas destrutivamente através desse nome da fila estão sujeitas à política do AMS.

Referências relacionadas

Exibir Gerenciador da Fila de Mensagens (DSPMQM)

Configurar a Política de segurança do MQM (SETMQMSPL)

IBM i *Testando uma política para o AMS no IBM i*

Use os aplicativos de amostra fornecidos com o produto para testar suas políticas de segurança.

Sobre esta tarefa

É possível usar os aplicativos de amostra fornecidos com o IBM MQ, tais como AMQSPUT4, AMQSGET4, AMQSGBR4 e ferramentas como WRKMQMMSG para colocar, procurar e obter mensagens usando o nome da fila PROTECTED.

Desde que tudo seja configurado corretamente, não deve haver diferença no comportamento do aplicativo para o de uma fila não protegida para este usuário.

Um usuário não configurado para Advanced Message Security ou um usuário que não tem a chave privada necessária para decriptografar a mensagem, contudo, não poderá visualizar a mensagem. O usuário recebe um código de conclusão de RCFAIL, equivalente a MQCC_FAILED (2) e o código de razão de RC2063 (MQRC_SECURITY_ERROR).

Para ver se a proteção AMS está em vigor coloque algumas mensagens de teste para a fila PROTECTED, por exemplo usando AMQSPUT0. É possível então criar uma fila de alias para procurar os dados brutos protegidos enquanto em repouso.

Procedimento

Para conceder as permissões necessárias para um usuário, execute:

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

Procurar usando o nome da fila ALIAS, por exemplo usando AMQSBCG4 ou WRKMQMMSG, deve revelar mensagens `scrambled` maiores em que uma procura da fila PROTECTED mostra mensagens de texto não criptografado.

As mensagens misturadas são visíveis, mas o texto não criptografado original não é decifrável usando a fila ALIAS, pois não há política para o AMS reforçar a correspondência a esse nome. Portanto, os dados brutos protegidos são retornados.

Referências relacionadas

Configurar a Política de segurança do MQM (SETMQMSPL)

Trabalhar com Mensagens MQ (WRKMQMMSG)

Eventos de comando e configuração para o AMS

Com o Advanced Message Security, é possível gerar mensagens de eventos de comando e configuração, que podem ser registradas e servir como um registro das mudanças da política para auditoria.

Os eventos de comando e configuração gerados pelo IBM MQ são mensagens do formato PCF enviadas para filas dedicadas no gerenciador de filas no qual o evento ocorre.

As mensagens de eventos de configuração são enviadas para a fila SYSTEM.ADMIN.CONFIG.EVENT.

As mensagens de eventos de comando são enviadas para a fila SYSTEM.ADMIN.COMMAND.EVENT.

Os eventos são gerados independentemente de ferramentas que você está usando para gerenciar as políticas de segurança do Advanced Message Security .

No Advanced Message Security, há quatro tipos de eventos gerados por diferentes ações em políticas de segurança:

- “Criando políticas de segurança no AMS” na página 687, que geram duas mensagens do evento do IBM MQ:
  - Um evento de configuração
  - Um evento de comando
- “Alterando Políticas de Segurança no AMS” na página 688, que gera três mensagens do evento do IBM MQ:
  - Um evento de configuração que contém os antigos valores de política de segurança
  - Um evento de configuração que contém os novos valores de política de segurança
  - Um evento de comando
- “Exibindo e fazendo dump de políticas de segurança no AMS” na página 688, que gera uma mensagem do evento do IBM MQ:
  - Um evento de comando
- “Removendo políticas de segurança no AMS” na página 690, que gera duas mensagens do evento do IBM MQ:
  - Um evento de configuração
  - Um evento de comando

Ativando e desativando a criação de log de eventos para o AMS

Você controla eventos de comando e configuração usando os atributos do gerenciador de filas **CONFIGEV** e **CMDEV**. Para ativar esses eventos, configure o atributo do gerenciador de filas apropriado para ENABLED. Para desativar esses eventos, configure o atributo apropriado do gerenciador de filas para DISABLED.

Procedimento

Eventos de Configuração

Para ativar eventos de configuração, configure **CONFIGEV** para ENABLED. Para desativar eventos de configuração, configure **CONFIGEV** para DISABLED. Por exemplo, é possível ativar eventos de configuração usando o comando MQSC a seguir:

```
ALTER QMGR CONFIGEV (ENABLED)
```

Eventos de Comando

Para ativar eventos de comando, configure **CMDEV** para ENABLED. Para ativar eventos de comandos, exceto comandos **DISPLAY MQSC** e Inquire PCF, configure **CMDEV** para NODISPLAY. Para desativar eventos de comando, configure **CMDEV** para DISABLED. Por exemplo, é possível ativar eventos de comandos usando o comando MQSC a seguir:

```
ALTER QMGR CMDEV (ENABLED)
```

Tarefas relacionadas

Controlando os eventos de configuração, de comando e de criador de logs no IBM MQ

Formato da mensagem do evento de comando para o AMS

A mensagem do evento de comando consiste em estrutura MQCFH e os parâmetros PCF a seguir.

Aqui estão os valores MQCFH selecionados:

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

Nota: O valor ParameterCount é dois porque sempre existem dois parâmetros do tipo MQCFGR (grupo). Cada grupo é constituído de parâmetros apropriados. Os dados do evento consistem em dois grupos, CommandContext e CommandData.

CommandContext contém:

EventUserID

Descrição :	O ID do usuário que emitiu o comando ou a chamada que gerou o evento. (Este é o mesmo ID do usuário que será usado para verificar a autoridade para emitir o comando ou chamada; para comandos recebidos de uma fila, este também será o identificador do usuário (UserIdentifier) do MD da mensagem de comando).
Identificador	MQCACF_EVENT_USER_ID.
Tipo de dado:	MQCFST.
Comprimento Máximo:	MQ_USER_ID_LENGTH.
Retornado:	Sempre.

EventOrigin

Descrição :	A origem da ação que causou o evento.
Identificador	MQIACF_EVENT_ORIGIN.
Tipo de dado:	MQCFIN.

Valores:

MQEVO_CONSOLE
Comando de console - linha de comandos.

MQEVO_MSG
Mensagem de comando do plug-in do IBM MQ Explorer.

Retornado: Sempre.

EventQMgr

Descrição : O gerenciador de filas no qual o comando ou chamada foi inserido. (O gerenciador de filas no qual o comando é executado e que gera o evento está no MD da mensagem do evento).

Identificador MQCACF_EVENT_Q_MGR.

Tipo de dado: MQCFST.

Comprimento Máximo: MQ_Q_MGR_NAME_LENGTH.

Retornado: Sempre.

EventAccountingToken

Descrição : Para comandos recebidos como uma mensagem (MQEVO_MSG), o token de conta (AccountingToken) do MD da mensagem de comando.

Identificador MQBACF_EVENT_ACCOUNTING_TOKEN.

Tipo de dado: MQCFBS.

Comprimento Máximo: MQ_ACCOUNTING_TOKEN_LENGTH.

Retornado: Somente se EventOrigin for MQEVO_MSG.

EventIdentityData

Descrição : Para comandos recebidos como uma mensagem (MQEVO_MSG), os dados de identificação do aplicativo (ApplIdentityData) do MD da mensagem de comando.

Identificador MQCACF_EVENT_APPL_IDENTITY.

Tipo de dado: MQCFST.

Comprimento Máximo: MQ_APPL_IDENTITY_DATA_LENGTH.

Retornado: Somente se EventOrigin for MQEVO_MSG.

EventApplType

Descrição : Para comandos recebidos como uma mensagem (MQEVO_MSG), o tipo do aplicativo (PutApplType) do MD da mensagem de comando.

Identificador MQIACF_EVENT_APPL_TYPE.

Tipo de dado: MQCFIN.

Retornado: Somente se EventOrigin for MQEVO_MSG.

EventApplName

Descrição : Para comandos recebidos como uma mensagem (MQEVO_MSG), o nome do aplicativo (PutApplName) do MD da mensagem de comando.

Identificador	MQCACF_EVENT_APPL_NAME.
Tipo de dado:	MQCFST.
Comprimento Máximo:	MQ_APPL_NAME_LENGTH.
Retornado:	Somente se EventOrigin for MQEVO_MSG.

EventApplOrigin

Descrição :	Para comandos recebidos como uma mensagem (MQEVO_MSG), os dados de origem do aplicativo (ApplOriginData) do MD da mensagem de comando.
Identificador	MQCACF_EVENT_APPL_ORIGIN.
Tipo de dado:	MQCFST.
Comprimento Máximo:	MQ_APPL_ORIGIN_DATA_LENGTH.
Retornado:	Somente se EventOrigin for MQEVO_MSG.

Comando:

Descrição :	O código de comando.
Identificador	MQIACF_COMMAND.
Tipo de dado:	MQCFIN.
Valores:	**MQCMD_INQUIRE_PROT_POLICY valor numérico 205** **MQCMD_CREATE_PROT_POLICY valor numérico 206** **MQCMD_DELETE_PROT_POLICY valor numérico 207** **MQCMD_CHANGE_PROT_POLICY valor numérico 208** Estes são definidos no IBM MQ 8.0 `cmqcfc.h`
Retornado:	Sempre.

CommandData contém elementos PCF que incluem o comando PCF.

Formato da mensagem do evento de configuração para o AMS

Os eventos de configuração são mensagens PCF de formato padrão do Advanced Message Security.

Valores possíveis para o descritor de mensagens MQMD podem ser localizados em Mensagem do evento MQMD (descritor de mensagens)

Aqui estão os valores MQMD selecionados:

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR           //for both CLI and command server
```

O buffer de mensagem consiste na estrutura MQCFH e a estrutura de parâmetros que a segue. Valores possíveis do MQCFH podem ser localizados em Mensagem do evento MQCFH (cabeçalho PCF).

Aqui estão os valores MQCFH selecionados:

```
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_OBJECT}
```

Os parâmetros seguindo MQCFH são:

EventUserID

Descrição : O ID do usuário que emitiu o comando ou a chamada que gerou o evento. (Este é o mesmo ID do usuário que será usado para verificar a autoridade para emitir o comando ou chamada; para comandos recebidos de uma fila, este também será o identificador do usuário (UserIdentifier) do MD da mensagem de comando).

Identificador **MQCACF_EVENT_USER_ID**

Tipo de dado: MQCFST.

Comprimento Máximo: MQ_USER_ID_LENGTH.

Retornado: Sempre.

SecurityId

Descrição : Valor de MQMD.AccountingToken no caso de mensagem do servidor de comandos ou SID do Windows para comando local.

Identificador **MQBACF_EVENT_SECURITY_ID**

Tipo de dado: MQCBS.

Comprimento Máximo: MQ_SECURITY_ID_LENGTH.

Retornado: Sempre.

EventOrigin

Descrição : A origem da ação que causou o evento.

Identificador **MQIACF_EVENT_ORIGIN**

Tipo de dado: MQCFIN.

Valores:

MQEVO_CONSOLE
Comando de console - linha de comandos.

MQEVO_MSG
Mensagem de comando a partir do plug-in do IBM MQ Explorer.

Retornado: Sempre.

EventQMgr

Descrição : O gerenciador de filas no qual o comando ou chamada foi inserido. (O gerenciador de filas no qual o comando é executado e que gera o evento está no MD da mensagem do evento).

Identificador **MQCACF_EVENT_Q_MGR**

Tipo de dado: MQCFST

Comprimento Máximo: MQ_Q_MGR_NAME_LENGTH

Retornado: Sempre.

ObjectType

Descrição : Tipo de objeto.

Identificador **MQIACF_OBJECT_TYPE**

Tipo de dado: MQCFIN

Valor: **MQOT_PROT_POLICY**
Política de proteção do Advanced Message Security. **1019** - um valor numérico definido no IBM MQ 8.0 ou no arquivo `cmqc.h`.

Retornado: Sempre.

PolicyName

Descrição : O nome da política do Advanced Message Security.

Identificador **MQCA_POLICY_NAME**.

Tipo de dado: MQCFST.

Valor: **2112** - um valor numérico definido no IBM MQ 8.0 ou no arquivo `cmqc.h`.

Comprimento Máximo: MQ_OBJECT_NAME_LENGTH.

Retornado: Sempre.

PolicyVersion

Descrição : A versão da política do Advanced Message Security.

Identificador **MQIA_POLICY_VERSION**

Tipo de dado: MQCFIN

Value **238** - um valor numérico definido no IBM MQ 8.0 ou no arquivo `cmqc.h` .

Retornado: Sempre

TolerateFlag

Descrição : O sinalizador de tolerância de política do Advanced Message Security.

Identificador **MQIA_TOLERATE_UNPROTECTED**

Tipo de dado: MQCFIN

Value **235** - um valor numérico definido no IBM MQ 8.0 ou no arquivo `cmqc.h`.

Retornado: Sempre.

SignatureAlgorithm

Descrição : O algoritmo de assinatura da política do Advanced Message Security.

Identificador **MQIA_SIGNATURE_ALGORITHM**

Tipo de dado: MQCFIN

Valor: **236** - um valor numérico definido no IBM MQ 8.0 ou no arquivo `cmqc.h`.

Retornado: Sempre que houver um algoritmo de assinatura definido na política do Advanced Message Security

EncryptionAlgorithm

Descrição : O algoritmo de criptografia da política do Advanced Message Security.

Identificador	**MQIA_ENCRYPTION_ALGORITHM**
Tipo de dado:	MQCFIN
Valor:	**237** - um valor numérico definido no IBM MQ 8.0 ou no arquivo `cmqc.h`.
Retornado:	Sempre que houver um algoritmo de criptografia definido na política do IBM MQ

SignerDNs

Descrição :	Assunto DistinguishedName dos assinantes permitidos.
Identificador	**MQCA_SIGNER_DN**
Tipo de dado:	MQCFSL
Valor:	**2113** - um valor numérico definido no IBM MQ 8.0 ou no arquivo `cmqc.h`.
Comprimento Máximo:	Maior DN de assinante na política, mas não maior que MQ_DISTINGUISHED_NAME_LENGTH
Retornado:	Sempre definido na política IBM MQ.

RecipientDNs

Descrição :	Assunto DistinguishedName dos assinantes permitidos.
Identificador	**MQCA_RECIPIENT_DN**
Tipo de dado:	MQCFSL
Valor:	**2114** - um valor numérico definido no IBM MQ 8.0 ou no arquivo `cmqc.h`.
Comprimento Máximo:	Maior DN do destinatário na política, mas não maior que MQ_DISTINGUISHED_NAME_LENGTH.
Retornado:	Sempre definido na política IBM MQ.

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte seu representante local do IBM para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja nenhum direito de propriedade intelectual da IBM. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou aplicativos de patentes pendentes relativas aos assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum sobre tais patentes. é possível enviar pedidos de licença, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
U.S.A.

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. São feitas alterações periódicas nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. IBM pode aperfeiçoar e/ou alterar no produto(s) e/ou programa(s) descritos nesta publicação a qualquer momento sem aviso prévio.

Todas as referências nessas informações a websites não IBM são fornecidas somente por conveniência e de forma alguma são um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Os licenciados deste programa que desejarem obter informações sobre este assunto com o propósito de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações trocadas, deverão entrar em contato com:

Av. Pasteur, 138-146
Av. Pasteur, 138-146

Botafogo
Rio de Janeiro, RJ
U.S.A.

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do IBM Customer Agreement, IBM Contrato de Licença do Programa Internacional ou qualquer contrato equivalente entre as partes.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disto, algumas medidas podem ter sido estimadas através de extrapolação. Os resultados reais podem variar. usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam somente metas e objetivos.

Essas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT:

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas.

Se estiver visualizando estas informações em formato eletrônico, as fotografias e ilustrações coloridas poderão não aparecer.

Informações sobre a Interface de Programação

As informações da interface de programação, se fornecidas, destinam-se a ajudá-lo a criar software aplicativo para uso com este programa.

Este manual contém informações sobre as interfaces de programação desejadas que permitem que o cliente grave programas para obter os serviços do IBM MQ

No entanto, estas informações também podem conter informações sobre diagnósticos, modificações e ajustes. As informações sobre diagnósticos, modificações e ajustes são fornecidas para ajudá-lo a depurar seu software aplicativo.

Importante: Não use essas informações de diagnóstico, modificação e ajuste como uma interface de programação, pois elas estão sujeitas a mudanças

Marcas comerciais

IBM, o logotipo IBM , ibm.com, são marcas registradas da IBM Corporation, registradas em várias jurisdições no mundo todo Uma lista atual de marcas registradas da IBM está disponível na Web em "Informações de copyright e marca registrada"www.ibm.com/legal/copytrade.shtml. Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas.

Microsoft e Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Este produto inclui software desenvolvido pelo Projeto Eclipse (https://www.eclipse.org/).

Java e todas as marcas registradas e logotipos baseados em Java são marcas ou marcas registradas da Oracle e/ou de suas afiliadas.

Part Number:

(1P) P/N: