9.4

Protección de IBM MQ

Nota

Antes de utilizar esta información y el producto al que da soporte, lea la información en "Avisos" en la página 707.

Esta edición se aplica a la versión 9 release 4 de IBM® MQ y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Cuando envía información a IBM, otorga a IBM un derecho no exclusivo para utilizar o distribuir la información de la forma que considere adecuada, sin incurrir por ello en ninguna obligación con el remitente.

© Copyright International Business Machines Corporation 2007, 2024.

Contenido

Protección de IBM MQ

La seguridad es una consideración importante para los desarrolladores de aplicaciones IBM MQ y para los administradores del sistema IBM MQ . Como mínimo, debe asegurarse de que todo el hardware y el software dentro de la zona segura y en las estaciones de trabajo del operador estén dentro de su ciclo de vida de soporte, estén actualizados con las actualizaciones de software obligatorias y se apliquen rápidamente las actualizaciones de seguridad.

Referencia relacionada

Gestión de vulnerabilidades de seguridad de IBM

z/OS IBM Z y LinuxOne Security Portal

Visión general de seguridad

Este colección de temas presentan los conceptos de seguridad de IBM MQ.

Primero se presentan los conceptos y mecanismos de seguridad, ya que se aplican a cualquier sistema, seguidos de un debate sobre los mecanismos de seguridad que se han implementado en IBM MQ.

Los aspectos de seguridad comúnmente aceptados son los siguientes:

- “Identificación y autenticación” en la página 7
- “Autorización” en la página 9
- “Auditoría” en la página 9
- “Confidencialidad” en la página 10
- “Integridad de datos” en la página 10

Los *mecanismos de seguridad* son herramientas técnicas y métodos técnicos que se utilizan para implementar los servicios de seguridad. Un mecanismo puede funcionar por sí solo, o con otros, para proporcionar un servicio determinado. Los siguientes son ejemplos de mecanismos de seguridad comunes:

- “Criptografía” en la página 11
- “Resúmenes de mensajes y firmas digitales” en la página 13
- “Certificados digitales” en la página 13
- “Infraestructura de claves públicas (PKI)” en la página 18

Cuando planifique una implementación de IBM MQ, considere qué mecanismos de seguridad necesita para implementar estos aspectos de seguridad que son importantes para usted. Para obtener información acerca de lo que ha de tener en cuenta después de que haya leído estos temas, consulte “Planificación de los requisitos de seguridad” en la página 90.

Identificación y autenticación

La *identificación* es la capacidad de identificar de forma exclusiva a un usuario de un sistema o una aplicación que se está ejecutando en el sistema. La *autenticación* es la capacidad de demostrar que un usuario o una aplicación es realmente quién dicha persona o aplicación asegura ser.

Por ejemplo, considere el caso de un usuario que se conecta a un sistema especificando un ID de usuario y una contraseña. El sistema utiliza el ID de usuario para identificar al usuario. El sistema autentica al usuario en el momento de la conexión comprobando que la contraseña proporcionada es correcta.

Identificación y autenticación en IBM MQ

Cuando una aplicación se conecta a IBM MQ, siempre se asocia una identidad de usuario con la conexión. La identidad de usuario es inicialmente el ID de usuario del sistema operativo que está asociado con

© Copyright IBM Corp. 2007, 2024

el proceso de aplicación. Esta identidad suele ser suficiente para las aplicaciones enlazadas localmente que están alojadas en el mismo sistema que el gestor de colas. Sin embargo, el gestor de colas también puede autenticar y modificar la identidad asociada con la conexión de varias maneras. La autenticación de la identidad asociada con una conexión es importante cuando las aplicaciones cliente que no son necesariamente fiables se conectan a un gestor de colas a través de una red.

La identidad que está asociada con una conexión de aplicación a un gestor de colas IBM MQ se puede establecer utilizando cualquiera de los mecanismos siguientes:

- Cuando una aplicación se conecta a un gestor de colas, puede proporcionar un ID de usuario y una contraseña. El gestor de colas valida las credenciales basándose en su configuración. Por ejemplo, el ID de usuario y la contraseña se pueden pasar al sistema operativo del gestor de colas, o al servidor LDAP, para autenticarse.
- V 9.4.0 A partir de IBM MQ 9.3.4, una aplicación también puede proporcionar una señal de autenticación que obtiene de un servidor de autenticación externo. Para obtener más información sobre las señales de autenticación, consulte “Cómo trabajar con señales de autenticación” en la página 331.
- Un canal de cliente se puede configurar para utilizar la autenticación mutua TLS, si se ha configurado con un certificado digital válido. La autenticación TLS se puede combinar con una regla de autenticación de canal (CHLAUTH) para asociar un ID de usuario adecuado con la conexión. Para obtener más información, consulte “Cómo proporciona TLS identificación, autenticación, confidencialidad e integridad” en la página 20,
- Las reglas de autenticación de canal (CHLAUTH) pueden alterar temporalmente la identidad basándose en la información sobre la conexión. Por ejemplo, una regla de autenticación de canal puede establecer el ID de usuario asociado a una conexión basándose en la dirección IP del cliente.
- El código de salida personalizado puede establecer una identidad basada en los criterios que elija.

La identidad y la autenticación también son aplicables a los canales entre dos gestores de colas. Estos canales se conocen como canales de mensajes. Cuando se inicia un canal de mensajes, el agente de canal de mensajes (MCA) en cada extremo del canal puede autenticar su asociado. Esta técnica se conoce como *autenticación mutua*. Para el MCA emisor, ofrece la garantía de que el asociado que está a punto de enviar los mensajes es auténtico. Del mismo modo, el MCA receptor está asegurado de que está a punto de recibir mensajes de un socio genuino.

Cuando se ha establecido una identidad y se ha autenticado si es necesario, IBM MQ la utiliza de varias maneras:

- Es importante destacar que, de forma predeterminada, las comprobaciones posteriores de “Autorización” en la página 9 se realizan utilizando esta identidad. Por ejemplo, si una aplicación intenta colocar un mensaje en una cola, el gestor de colas confirma que la identidad asociada con la aplicación tiene autorización 'put' en el objeto de cola.
- Además, cada mensaje puede contener información de *contexto de mensaje* . Esta información se encuentra en el descriptor de mensaje (MQMD). El gestor de colas puede generar automáticamente el contexto de mensaje cuando una aplicación transfiere el mensaje a una cola. De forma alternativa, la aplicación puede proporcionar el contexto de mensaje si el ID de usuario asociado a la aplicación está autorizado para hacerlo. Esta información de contexto en un mensaje proporciona a la aplicación que recibe la información de mensaje sobre el originador del mensaje. Por ejemplo, contiene el nombre de la aplicación que ha transferido el mensaje y el ID de usuario asociado a la aplicación.

No rechazo

El objetivo general del servicio contra rechazos es poder demostrar que un mensaje concreto está asociado a un individuo concreto.

El *servicio contra rechazos* se puede considerar una ampliación del servicio de identificación y autenticación. En general, el servicio contra rechazos se aplica cuando se transmiten electrónicamente los datos; por ejemplo, un pedido a un intermediario de bolsa para comprar o vender acciones o una orden de transferencia a un banco de una cuenta a otra.

El servicio contra rechazos puede contener más de un componente y cada componente proporciona una función diferente. Si el emisor de un mensaje niega alguna vez haberlo enviado, el servicio contra rechazos con *prueba de origen* puede proporcionar al receptor una prueba irrefutable de que el mensaje lo ha enviado esta persona concreta. Si el receptor de un mensaje niega alguna vez haberlo recibido, el servicio contra rechazos con *prueba de entrega* puede proporcionar al emisor una prueba irrefutable de que el mensaje ha sido recibido por esta persona concreta.

En la práctica, obtener una prueba con una seguridad prácticamente del 100% o una prueba irrefutable, es un objetivo difícil de alcanzar. En el mundo real, nada es absolutamente seguro. Gestionar la seguridad está más relacionado con gestionar los riesgos a un nivel que resulte aceptable para la empresa. En este tipo de entornos, la expectativa más realista del servicio contra rechazos es poder proporcionar una prueba que resulte admisible y apoye la causa ante los tribunales.

El servicio contra rechazos es un servicio de seguridad importante en un entorno IBM MQ ya que IBM MQ es un medio de transmitir datos electrónicamente. Por ejemplo, es posible que necesite una prueba actual de que un mensaje determinado lo ha enviado o recibido una aplicación asociada una persona concreta.

IBM MQ con Advanced Message Security no proporciona un servicio contra rechazos como parte de su función básica. No obstante, esta documentación de producto contiene sugerencias sobre cómo puede proporcionar su propio servicio contra rechazos en un entorno IBM MQ escribiendo sus propios programas de salida.

Autorización

La *autorización* protege los recursos importantes de un sistema, ya que limita el acceso solamente a los usuarios autorizados y a sus aplicaciones. Impide que los recursos se utilicen sin la autorización necesaria.

Autorización en IBM MQ

Puede utilizar la autorización para limitar lo que determinados individuos o aplicaciones pueden hacer en el entorno de IBM MQ.

A continuación se muestran algunos ejemplos de autorización en un entorno de IBM MQ:

- Permitir solamente que el administrador autorizado pueda emitir mandatos para gestionar recursos de IBM MQ.
- Permitir que una aplicación se conecte a un gestor de colas solamente si el ID de usuario asociado a la aplicación tiene autorización para hacerlo.
- Permitir que una aplicación abra solamente las colas que sean necesarias para su funcionamiento.
- Permitir que una aplicación se suscriba solamente a los temas que sean necesarios para su funcionamiento.
- Permitir que una aplicación realice en una cola solamente las operaciones que sean necesarias para su funcionamiento. Por ejemplo, es posible que una aplicación sólo necesite examinar los mensajes de una cola determinada y no necesite transferir ni obtener mensajes.

Para obtener más información sobre cómo configurar la autorización, consulte “Planificación de la autorización” en la página 94 y los subtemas asociados.

Auditoría

La *auditoría* es el proceso de registrar y comprobar sucesos para detectar si ha tenido lugar una actividad no esperada o no autorizada, o si se ha llevado a cabo algún intento para realizar dicha actividad.

Auditoría en IBM MQ

IBM MQ puede emitir mensajes de sucesos para registrar que ha tenido lugar actividad poco usual.

A continuación se muestran algunos ejemplos de auditoría en un entorno de IBM MQ:

- Una aplicación intenta abrir una cola que no tiene autorización para abrir. Se emite un mensaje de suceso de instrumentación. Al inspeccionar el mensaje de suceso, descubre que se ha producido este intento y puede decidir qué acción es necesaria.
- Una aplicación intenta abrir un canal, pero el intento falla porque la conexión TLS no está permitida. Se emite un mensaje de suceso de instrumentación. Al inspeccionar el mensaje de suceso, descubre que se ha producido este intento y puede decidir qué acción es necesaria.

Confidencialidad

El servicio de *confidencialidad* protege la información confidencial para que no pueda divulgarse sin la autorización correspondiente.

Cuando los datos confidenciales se almacenan localmente, los mecanismos de control de accesos pueden ser suficientes para protegerlos basándose en la suposición de que no pueden leerse los datos si no se puede acceder a los mismos. Si se necesita un nivel de seguridad mayor, los datos se pueden cifrar.

Cifre los datos confidenciales cuando se transmitan a través de una red de comunicaciones, especialmente una red insegura, como por ejemplo Internet. En un entorno de red, los mecanismos de control de accesos no son una protección eficaz contra los intentos de interceptar los datos, como por ejemplo, las escuchas telefónicas ilegales.

Confidencialidad en IBM MQ

Puede implementar la confidencialidad en IBM MQ cifrando mensajes.

La confidencialidad puede asegurarse en un entorno IBM MQ como se indica a continuación:

- Después de que un MCA emisor obtenga un mensaje de una cola de transmisión, IBM MQ utiliza TLS para cifrar el mensaje antes de enviarlo a través de la red al MCA receptor. En el otro extremo del canal, el mensaje se descifra antes de que el MCA receptor lo transfiera a la cola de destino.
- Mientras que los mensajes se almacenan en una cola local, los mecanismos de control de accesos proporcionados por IBM MQ se podrían considerar suficientes para proteger su contenido contra una revelación no autorizada. Sin embargo, para un mayor nivel de seguridad, puede utilizar Advanced Message Security para cifrar los mensajes almacenados en las colas.
- z/OS Los mensajes almacenados en las colas locales se pueden cifrar en reposo utilizando el cifrado de conjunto de datos de z/OS.

  Consulte la sección Confidencialidad para los datos en reposo en IBM MQ for z/OS con cifrado de conjunto de datos. para obtener más información.

Integridad de datos

El servicio de *integridad de datos* detecta si se han modificado los datos de forma no autorizada.

Hay dos modos de alterar los datos: de forma accidental, mediante errores de hardware y transmisión o debido a un ataque deliberado. Muchos productos de hardware y protocolos de transmisión disponen de mecanismos para detectar y corregir los errores de hardware y transmisión. La finalidad del servicio de integridad de datos es detectar un ataque deliberado.

El único objetivo del servicio de integridad de datos es detectar si se han modificado los datos. Su objetivo no es restaurar los datos a su estado original si se han modificado.

Los mecanismos de control de accesos pueden ayudar a la integridad de los datos, dado que los datos no se pueden modificar si se deniega el acceso. Pero, del mismo modo que ocurre con la confidencialidad, los mecanismos de control de accesos no resultan eficaces en un entorno de red.

Integridad de los datos en IBM MQ

La integridad de datos puede asegurarse en un entorno IBM MQ como se indica a continuación:

- Puede utilizar TLS para detectar si el contenido de un mensaje se ha modificado de forma deliberada mientras se transmitía a través de una red. En TLS, el algoritmo de resumen de mensaje proporciona la detección de mensajes modificados en tránsito.

  Todas las CipherSpecs de IBM MQ proporcionan un algoritmo de resumen de mensaje, excepto para TLS_RSA_WITH_NULL_NULL, que no proporciona integridad de los datos del mensaje.

  IBM MQ detecta los mensajes modificados al recibirlos; al recibir un mensaje modificado, IBM MQ se graba un mensaje de error AMQ9661 en el registro de errores y el canal se detiene.
- Mientras los mensajes se almacenan en una cola local, los mecanismos de control de accesos que proporciona IBM MQ pueden considerarse suficientes para impedir la modificación deliberada del contenido de los mensajes.

  Sin embargo, para un mayor nivel de seguridad, puede utilizar Advanced Message Security para detectar si el contenido de un mensaje se ha modificado deliberadamente entre la hora cuando se colocó el mensaje en la cola y la hora cuando se recuperó de la cola.

  Si se detecta un mensaje modificado, la aplicación que intenta recibir el mensaje recibe un código de retorno MQRC_SECURITY_ERROR (2063). Si la aplicación utiliza una llamada MQGET , el mensaje también se mueve a SYSTEM.PROTECTION.ERROR.QUEUE .

Conceptos de cifrado

En esta colección de temas se describen los conceptos de cifrado aplicables a IBM MQ.

El término *entidad* se utiliza para hacer referencia a un gestor de colas, un IBM MQ MQI client, un usuario individual o cualquier otro sistema capaz de intercambiar mensajes.

Criptografía

El cifrado es el proceso de convertir texto legible, denominado *texto plano*, en un formato ilegible, denominado *texto cifrado*.

Esto se produce como se indica a continuación:

1. El emisor convierte el mensaje de texto plano en texto cifrado. Esta parte del proceso se denomina *cifrado* (algunas veces, se denomina *codificación*).
2. El texto cifrado se transmite al receptor.
3. El receptor vuelve a convertir el mensaje de texto cifrado en su formato en texto plano. Esta parte del proceso se denomina *descifrado* (algunas veces, *decodificación*).

La conversión requiere una secuencia de operaciones matemáticas que cambian el aspecto del mensaje durante la transmisión pero no afecta el contenido. Las técnicas de cifrado pueden garantizar la confidencialidad y proteger los mensajes contra la visualización no autorizada (escuchas secretas), ya que un mensaje cifrado no es inteligible. Las firmas digitales, que ofrecen una garantía de la integridad del mensaje, utilizan técnicas de cifrado. Consulte “Firmas digitales en SSL/TLS” en la página 23 para obtener más información.

Las técnicas de cifrado requieren un algoritmo general, que pasa a ser específico mediante el uso de claves. Hay dos clases de algoritmos:

- Los que requieren que ambas partes utilicen la misma clave secreta. Los algoritmos que utilizan una clave compartida se conocen como algoritmos *simétricos*. La Figura 1 en la página 12 ilustra el cifrado de claves simétricas.
- Las que utilizan una clave para cifrado y una clave diferente para descifrado. Una de estas debe mantenerse secreta pero la otra puede ser pública. Los algoritmos que utilizan los pares de claves pública y privada se conocen como algoritmos *simétricos*. La Figura 2 en la página 12 ilustra el cifrado de claves asimétricas, que también se conoce también como *cifrado de claves públicas*.

Los algoritmos de cifrado y descifrado utilizados pueden ser públicos pero la clave secreta compartida y la clave privada debe mantenerse secreta.

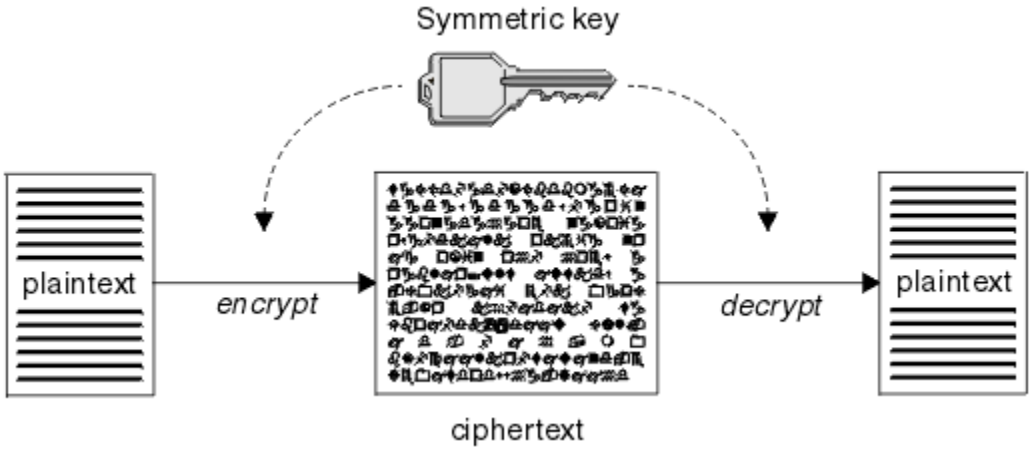


Figura 1. Cifrado de claves simétricas

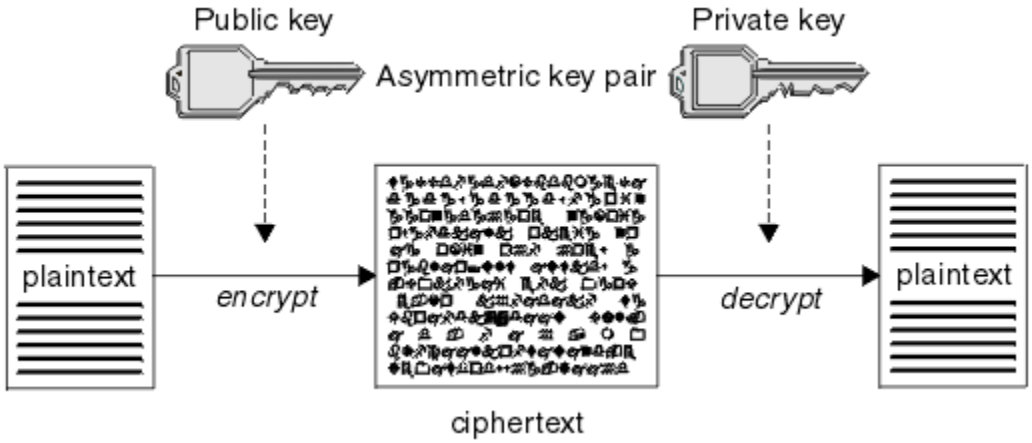


Figura 2. Cifrado de claves asimétricas

La Figura 2 en la página 12 muestra texto plano cifrado con la clave pública del receptor y descifrado con la clave privada del receptor. Solamente el receptor al que va destinado tiene la clave privada para descifrar el texto cifrado. Tenga en cuenta que el emisor también puede cifrar mensajes con una clave privada, con lo que cualquiera que tenga la clave pública del emisor puede descifrar el mensaje, con la seguridad de que el mensaje procede del emisor.

Con los algoritmos asimétricos, los mensajes se cifran o con la clave pública o con la clave privada pero solamente se pueden descifrar con la otra clave. Solamente la clave privada es secreta, la clave pública la puede conocer cualquiera. Con los algoritmos simétricos, la clave compartida solamente deben conocerla las dos partes. Esto se denomina el *problema de distribución de claves*. Los algoritmos simétricos son más lentos pero tienen la ventaja de que no existe el problema de distribución de claves.

Otra terminología asociada al cifrado es:

Potencia
La potencia del cifrado la determina el tamaño de las claves. Los algoritmos asimétricos requieren claves grandes, por ejemplo:

1024 bits	Clave asimétrica de potencia baja
2048 bits	Clave asimétrica de potencia media
4096 bits	Clave asimétrica de potencia alta

Las claves asimétricas son más pequeñas: las claves de 256 bits le proporcionan un cifrado muy potente.

Algoritmo de cifrado de bloques
Estos algoritmos cifran los datos por bloques. Por ejemplo, el algoritmo RC2 de RSA Data Security Inc. utiliza bloques de 8 bytes de longitud. Los algoritmos de bloques normalmente son más lentos que los algoritmos de flujo.

Algoritmo de cifrado de flujo
Estos algoritmos funcionan en cada byte de datos. Los algoritmos de flujo normalmente son más rápidos que los algoritmos de bloques.

Resúmenes de mensajes y firmas digitales

Un resumen de mensaje es una representación numérica de tamaño fijo del contenido de un mensaje. El resumen del mensaje se calcula mediante una función hash y se puede cifrar, formando una firma digital.

La función hash se utiliza para calcular si un resumen de mensaje cumple con dos criterios:

- Debe ser unidireccional. No debe ser posible invertir la función para encontrar el mensaje correspondiente a un resumen de mensaje específico mediante otros medios que no sea la comprobación de todos los mensajes posibles.
- Debe ser matemáticamente imposible encontrar dos mensajes cuyo valor hash sean iguales al mismo resumen.

El resumen de mensaje se envía con el mensaje propiamente dicho. El receptor puede generar un resumen para el mensaje y compararlo con el resumen del emisor. La integridad del mensaje se verifica cuando los dos resúmenes de mensaje son iguales. Si el mensaje ha sido manipulado de algún modo durante la transmisión, es prácticamente seguro que el resultado sería un resumen de mensaje diferente.

Un resumen de mensaje creado utilizando una clave simétrica secreta es conocido como un Código de Autenticación de Mensaje (MAC), ya que puede garantizar que el mensaje no se ha modificado.

El emisor también puede generar un resumen de mensaje y luego cifrar el resumen utilizando la clave privada de un par de claves asimétricas, formando una firma digital. El receptor debe descifrar luego la firma, antes de compararla con un resumen generado localmente.

Conceptos relacionados

“Firmas digitales en SSL/TLS” en la página 23
Una firma digital se crea cifrando una representación de un mensaje. El cifrado utiliza la clave privada del que firma y, por motivos prácticos, suele operar en un resumen del mensaje en lugar de hacerlo en el mensaje propiamente dicho.

Certificados digitales

Los certificados digitales protegen contra la suplantación de identidad, certificando que una clave pública pertenece a una entidad especificada. Son emitidos por una Entidad emisora de certificados.

Los certificados digitales protegen contra la suplantación de identidad, ya que un certificado digital enlaza una clave pública con su propietario, tanto si el propietario es un usuario, un gestor de colas o cualquier otro tipo de entidad. Los certificados digitales también se denominan certificados de claves públicas ya que le garantizan la propiedad de una clave pública cuando utiliza un esquema de claves asimétrico. Un certificado digital contiene la clave pública para una entidad y es una declaración de que la clave pública pertenece a dicha entidad:

- Cuando el certificado es para una entidad individual, el certificado se denomina *certificado personal* o *certificado de usuario*.
- Cuando el certificado es para una Entidad emisora de certificados, el certificado se denomina *certificado CA* o *certificado de firmante*.

Si las claves públicas las envía directamente su propietario a otra entidad, existe el riesgo de que el mensaje pueda ser interceptado y de que la clave pública sea sustituida por otra. Esto se conoce como *interposición de intrusos*. La solución a este problema es intercambiar las claves públicas mediante una entidad de terceros fiable, con lo que obtiene una mayor garantía de que la clave pública realmente pertenece a la entidad con la que se está comunicando. En lugar de enviar directamente la clave

pública, se solicita a la entidad de terceros fiable que la incorpore en un certificado digital. El tercero de confianza que emite certificados digitales se llama autoridad de certificación (CA), tal como se describe en "Entidades emisoras de certificados" en la página 15.

Qué es un certificado digital

Los certificados digitales contienen elementos de información específicos, según se determina en el estándar X.509.

Los certificados digitales que utiliza IBM MQ se ajustan al estándar X.509, que especifica la información necesaria y el formato con que se envía. X.509 es la parte de la infraestructura de autenticación de las series de estándares X.500.

Los certificados digitales contienen como mínimo la información siguiente acerca de la entidad que se está certificando:

- La clave pública del propietario
- El nombre distinguido del propietario
- El Nombre distinguido de la CA que ha emitido el certificado
- La fecha a partir de la cual es válido el certificado
- La fecha de caducidad del certificado
- El número de versión del formato de datos del certificado como se define en x.509. La versión actual del estándar x.509 es la Versión 3, y la mayoría de los certificados se ajustan a dicha versión.
- Un número de serie. Se trata de un identificador exclusivo asignado por la CA que emitió el certificado. El número de serie es exclusivo dentro de la CA que emitió el certificado: no hay dos certificados firmados por el mismo certificado de CA que tengan el mismo número de serie.

Un certificado X.509 Versión 2 también contiene un Identificador de emisor y un Identificador de sujeto, y un certificado X.509 Versión 3 puede contener varias extensiones. Algunas extensiones de certificados, como por ejemplo la extensión de restricción básica, son *estándar* pero otras son específicas de la implementación. Una extensión puede ser *crítica*, en cuyo caso debe haber un sistema disponible para reconocer el campo; si no reconoce el campo, deberá rechazar el certificado. Si una extensión no es crítica, el sistema podrá omitirla si no la reconoce.

La firma digital de un certificado personal se genera utilizando la clave privada de la CA que ha firmado dicho certificado. Cualquier persona que necesite verificar el certificado personal puede utilizar la clave pública de la CA para hacerlo. El certificado de la CA contiene la clave pública.

Los certificados digitales no contienen la clave privada. Debe mantener la clave privada en secreto.

Requisitos para los certificados personales

IBM MQ da soporte a certificados digitales que cumplan con el estándar X.509. Requiere la opción de autenticación de cliente.

Dado que IBM MQ es un sistema de igual a igual, se considera como una autenticación de cliente en la terminología de SSL/TLS. Por consiguiente, cualquier certificado personal utilizado para la autenticación SSL/TLS debe permitir un uso de claves de autenticación de cliente. No todos los certificados de servidor tienen esta opción habilitada, por lo que es posible que el proveedor de certificados tenga que habilitar la autenticación de cliente en la CA raíz para un certificado seguro.

Además de los estándares que especifican el formato de datos para un certificado digital, también existen los estándares para determinar si un certificado es válido. Estos estándares se han actualizado a lo largo del tiempo para impedir ciertos tipos de infracción de seguridad. Por ejemplo, los certificados X.509 anteriores de la versión 1 y 2 no indicaban si el certificado se podía utilizar legítimamente para firmar otros certificados. Por lo tanto, era posible que un usuario malicioso obtuviese un certificado personal de un origen legítimo y crease nuevos certificados diseñados para suplantar a otros usuarios.

Al utilizar certificados X.509 de la versión 3, las extensiones de certificado BasicConstraints y KeyUsage se utilizan para especificar qué certificados pueden firmar legítimamente otros certificados. El estándar IETF RFC 5280 especifica una serie de reglas de validación de certificados que el software de aplicación

compatible debe implementar para evitar ataques de suplantación. Un conjunto de reglas de certificado se conoce como una política de validación de certificados.

Para obtener más información sobre las políticas de validación de certificados en IBM MQ, consulte “Políticas de validación de certificados en IBM MQ” en la página 47.

Entidades emisoras de certificados

Una Entidad emisora de certificados (CA) es una entidad de terceros fiable que emite certificados digitales que le garantizan que la clave pública de una entidad pertenece realmente a dicha entidad.

Las funciones de una CA son:

- Al recibir una solicitud de un certificado digital, verificar la identidad del solicitante antes de crear, firmar y devolver el certificado personal.
- Proporcionar la clave pública propia de la CA en su certificado de CA.
- Publicar listas de certificados que ya no son fiables en la Lista de revocación de certificados (CRL). Para obtener más información, consulte “Trabajar con certificados revocados” en la página 346
- Proporcionar acceso al estado de revocación del certificado utilizando un servidor de programa de respuesta OCSP

Nombres distinguidos

El nombre distinguido (DN) identifica de forma exclusiva una entidad en un certificado X.509.

Atención: En un filtro SSLPEER solo pueden utilizarse los atributos de la tabla siguiente. Los nombres distinguidos de certificado pueden contener otros atributos, pero no se permite filtrar en estos atributos.

Tabla 1. Los tipos de atributo encontrados en el nombre distinguido que se pueden utilizar en un filtro SSLPEER

Tipo de atributo	**Descripción**
SERIALNUMBER	Número de serie de certificado
MAIL	Dirección de correo electrónico
Deprecated E	Dirección de correo electrónico (En desuso por ser preferible MAIL)
UID o USERID	Identificador de usuario
CN	Nombre común
T	Título
OU	Nombre de la unidad organizativa
DC	Componente de dominio
O	Nombre de la organización
CALLE	Calle / Primera línea de dirección
L	Nombre de la localidad
ST (o SP o S)	Nombre del estado o provincia
PC	Código postal
C	País
UNSTRUCTUREDNAME	Nombre de host
UNSTRUCTUREDADDRESS	Dirección IP
DNQ	Calificador de nombre distinguido

El estándar X.509 define otros atributos que generalmente no forman parte del DN pero que pueden proporcionar extensiones opcionales al certificado digital.

El estándar X.509 proporciona un DN que se especifica con un formato de serie. Por ejemplo:

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

El Nombre común (CN) puede describir un usuario individual o cualquier otra entidad, por ejemplo un servidor web.

El DN puede contener varios atributos OU y DC. Sólo se permite una instancia de cada uno de los otros atributos. El orden de las entradas OU es importante: el orden especifica una jerarquía de nombres de unidades organizativas, con el nivel de unidad del más alto nivel en primer lugar. El orden de las entradas DC también es importante.

IBM MQ tolera ciertos nombres distinguidos (DN) malformados. Para obtener más información, consulte Reglas de IBM MQ para valores de SSLPEER.

Conceptos relacionados

"Qué es un certificado digital" en la página 14
Los certificados digitales contienen elementos de información específicos, según se determina en el estándar X.509.

Obtención de certificados personales de una entidad emisora de certificados

Puede obtener un certificado de una entidad emisora de certificados (CA) externa.

Un certificado digital se obtiene enviando información a un CA, en forma de una solicitud de certificado. El estándar X.509 define un formato para esta información, pero algunas CA tienen su propio formato. Las solicitudes de certificados las generan normalmente la herramienta de gestión de certificados que el sistema utiliza; por ejemplo:

- ALW Los mandatos **runmqakm** y V 9.4.0 V 9.4.0 **runmqktool** en AIX, Linux, and Windows.
- z/OS RACF en z/OS.

La información contiene el Nombre distinguido y la clave pública. Cuando la herramienta de gestión de certificados genera la solicitud de certificado, también genera la clave privada, que debe mantener en un lugar seguro. No distribuya nunca su clave privada.

Cuando la CA recibe la solicitud, la autorización comprueba su identidad antes de crear el certificado y devolverlo como un certificado personal.

La Figura 3 en la página 16 ilustra el proceso de obtener un certificado digital de una CA.

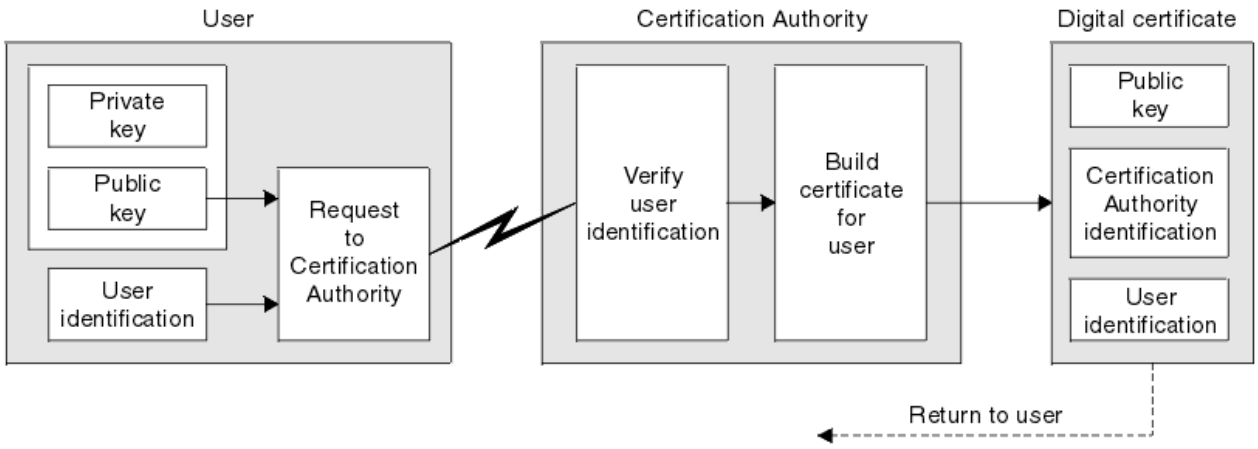


Figura 3. Obtención de un certificado digital

En el diagrama:

- La identificación de usuario incluye su Nombre distinguido de sujeto.
- La identificación de autoridad de certificación incluye el Nombre distinguido de la CA que está emitiendo el certificado.

Los certificados digitales contienen campos adicionales distintas de las que aparecen en el diagrama. Para obtener más información sobre el resto de campos en un certificado digital, consulte "Qué es un certificado digital" en la página 14.

Cómo funcionan las cadenas de certificados

Cuando recibe el certificado de otra entidad, es posible que necesite utilizar una *cadena de certificados* para obtener el certificado de la *CA raíz*.

La cadena de certificados, que también se conoce como la *vía de acceso de certificación*, es una lista de certificados que se utiliza para autenticar una entidad. La cadena, o vía de acceso, comienza por el certificado de esta entidad y cada uno de los certificados de la cadena lo forma la entidad identificada mediante el certificado siguiente de la cadena. La cadena termina con un certificado de la CA raíz. El certificado de la CA raíz siempre está firmado por la propia entidad emisora de certificados (CA). Las firmas de todos los certificados de la cadena se deben verificar hasta que se alcance el certificado de CA raíz.

La Figura 4 en la página 17 ilustra una vía de acceso de certificación desde el propietario del certificado a la CA raíz, donde la cadena de confianza comienza.

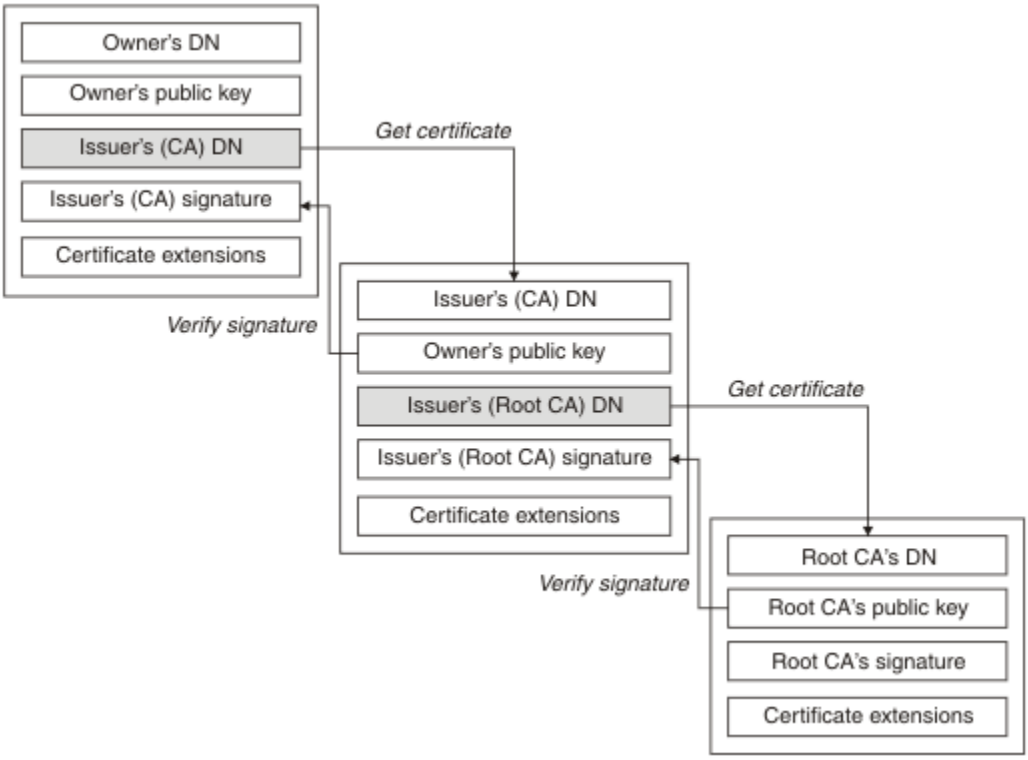


Figura 4. Cadena fiable

Cada certificado puede contener una o varias extensiones. Un certificado que pertenece a una CA contiene normalmente una extensión BasicConstraints con el distintivo isCA establecido para indicar que está permitido firmar otros certificados.

Cuando los certificados ya no son válidos

Los certificados digitales pueden caducar o revocarse.

Los certificados digitales se emiten durante un período fijo de tiempo y no son válidos después de su fecha de caducidad.

Los certificados se pueden revocar por varios motivos, entre ellos:

- El propietario ha cambiado a una organización distinta.
- La clave privada ya no es secreta.

IBM MQ puede comprobar si un certificado se ha revocado enviando una solicitud a un respondedor OCSP (Online Certificate Status Protocol) (en AIX, Linux, and Windows solamente). Alternativamente, pueden acceder a una CRL (Certificate Revocation List) en un servidor LDAP. La información de revocación OCSP y de CRL la publica una entidad emisora de certificados. Para obtener más información, consulte "Trabajar con certificados revocados" en la página 346.

Infraestructura de claves públicas (PKI)

Una infraestructura de claves públicas (PKI) es un sistema de recursos, políticas y servicios que da soporte al uso del cifrado de claves públicas para autenticar a las partes que participan en una transacción.

No hay ningún estándar individual que defina los componentes de una Infraestructura de clave pública, pero normalmente un PKI consta de entidades emisoras de certificados (CA) y entidades emisoras de registro (RA). Las CA proporcionan los servicios siguientes:

- Emisión de certificados digitales
- Validación de certificados digitales
- Revocación de certificados digitales
- Distribución de claves públicas

El estándar X.509 proporcionan la base para la industria estándar de la infraestructura Public Key Infrastructure.

Consulte "Certificados digitales" en la página 13 para obtener más información sobre los certificados digitales y las entidades emisoras de certificados (CA). Las RA verifican la información que se proporciona cuando se solicitan certificados digitales. Si la RA verifica esta información, la CA puede emitir un certificado digital para el solicitante.

Una PKI también puede proporcionar las herramientas para gestionar los certificados digitales y las claves públicas. Una PKI se describe a veces como una *jerarquía fiable* para la gestión de certificados digitales, aunque la mayor parte de las definiciones incluyen servicios adicionales. Algunas definiciones incluyen servicios de cifrado y firma digital, pero estos servicios no son esenciales para el funcionamiento de una PKI.

Protocolos de seguridad de cifrado: TLS

Los protocolos de cifrado proporcionan conexiones seguras, permitiendo que dos partes se comuniquen con privacidad e integridad de datos. El protocolo TLS (Transport Layer Security) ha evolucionado a partir del protocolo SSL (Secure Sockets Layer). IBM MQ ofrece soporte para TLS.

Los principales objetivos de ambos protocolos consisten en proporcionar confidencialidad, (que a veces recibe el nombre de *privacidad*), integridad de datos, identificación y autenticación utilizando certificados digitales.

Aunque los dos protocolos son parecidos, las diferencias son suficientemente significativas como para que SSL 3.0 y las diversas versiones de TLS no puedan interactuar.

Conceptos relacionados

"Protocolos de seguridad TLS en IBM MQ" en la página 25
IBM MQ da soporte al protocolo TLS (seguridad de la capa de transporte) para proporcionar seguridad a nivel de enlace para los canales de mensajes y los canales MQI.

Conceptos de TLS (Transport Layer Security)

El protocolo TLS permite que dos partes se identifiquen y autentiquen entre sí y se comuniquen con confidencialidad e integridad de datos. El protocolo TLS ha evolucionado a partir del protocolo Netscape SSL 3.0, pero TLS y SSL no pueden interactuar.

El protocolo TLS proporciona a las comunicaciones seguridad en Internet y permiten a las aplicaciones cliente/servidor comunicarse de una forma que es confidencial y fiable. Los protocolos tienen dos capas: un protocolo de registro y un protocolo de reconocimiento y éstos están en capas por encima de un protocolo de transporte como, por ejemplo, TCP/IP. Ambos utilizan técnicas de cifrado simétrico y asimétrico.

Una aplicación inicia una conexión TLS, que se convierte en el cliente TLS. La aplicación que recibe la conexión pasa a ser el servidor TLS. Cada nueva sesión se inicia con un reconocimiento, tal como lo define el protocolo TLS.

Se proporciona una lista completa de las CipherSpecs soportadas por IBM MQ en "Habilitación de CipherSpecs" en la página 428.

Para obtener más información sobre el protocolo SSL, consulte la información proporcionada en https://developer.mozilla.org/docs/Mozilla/Projects/NSS. Para obtener más información sobre el protocolo TLS, consulte la información proporcionada por el grupo de trabajo TLS en el sitio web de Internet Engineering Task Force en https://www.ietf.org

Visión general del reconocimiento SSL/TLS

El reconocimiento SSL/TLS permite que el cliente y el servidor TLS establezcan las claves secretas con las que se comunican.

Esta sección proporciona un resumen de los pasos que permiten que el cliente y el servidor TLS se comuniquen entre sí.

- Acordar la versión del protocolo que se va a utilizar.
- Seleccionar los algoritmos de cifrado.
- Autenticarse mutuamente intercambiando y validando certificados digitales.
- Utilizar técnicas de cifrado asimétrico para generar una clave secreta compartida, que evita el problema de distribución de claves. TLS utiliza entonces la clave compartida para el cifrado simétrico de los mensajes, lo cual es más rápido que el cifrado asimétrico.

Para obtener más información acerca de los algoritmos de cifrado y los certificados digitales, consulte la información relacionada.

En general, los pasos que se realizan durante el reconocimiento TLS son los siguientes:

1. El cliente TLS envía un mensaje de "saludo del cliente" que lista la información de cifrado como, por ejemplo, la versión de TLS y, según el orden de preferencias del cliente, las CipherSuites que soporta el cliente. El mensaje también contiene un serie de bytes aleatorios que se utilizan en cálculos posteriores. El protocolo permite que el mensaje de "saludo del cliente" incluya los métodos de compresión de datos soportados por el cliente.
2. El servidor TLS responde con un mensaje de "saludo del servidor" que contiene la CipherSuite elegida por el servidor en la lista que ha proporcionado el cliente, el ID de sesión y otra serie de bytes aleatorios. El servidor también envía su certificado digital. Si el servidor requiere un certificado digital para la autenticación del cliente, el servidor envía una "solicitud de certificado de cliente" que incluye una lista de los tipos de certificados soportados y los nombres distinguidos de las Autoridades de certificación (CA) aceptables.
3. El cliente TLS verifica el certificado digital del servidor. Para obtener más información, consulte "Cómo proporciona TLS identificación, autenticación, confidencialidad e integridad" en la página 20.
4. El cliente TLS envía la serie de bytes aleatorios que permite que tanto el cliente como el servidor calculen la clave secreta que se utilizará para cifrar los datos del mensaje posterior. La serie de bytes aleatorios se cifra con la clave pública del servidor.

5. Si el servidor TLS ha enviado una "solicitud de certificado de cliente", el cliente envía una serie de bytes aleatorios cifrada con la clave privada del cliente, junto con el certificado digital del cliente, o una "alerta que indica que no hay certificado digital". Esta alerta es simplemente un aviso, pero en algunas implementaciones el reconocimiento no se ejecuta correctamente si la autenticación de cliente es obligatoria.
6. El servidor TLS verifica el certificado del cliente. Para obtener más información, consulte “Cómo proporciona TLS identificación, autenticación, confidencialidad e integridad” en la página 20.
7. El cliente TLS envía al servidor un mensaje de "finalizado", que se cifra con la clave secreta, que indica que la parte de cliente del reconocimiento se ha completado.
8. El servidor TLS envía al cliente un mensaje de "finalizado", que se cifra con la clave secreta, que indica que la parte de servidor del reconocimiento se ha completado.
9. Durante la sesión TLS, el servidor y el cliente podrán intercambiar mensajes que estén cifrados simétricamente con la clave secreta compartida.

La Figura 5 en la página 20 ilustra el reconocimiento TLS.

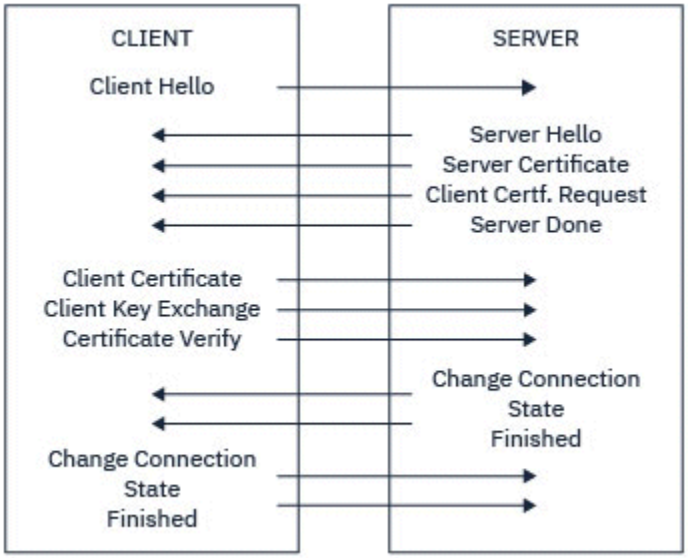


Figura 5. Visión general del reconocimiento TLS

Cómo proporciona TLS identificación, autenticación, confidencialidad e integridad

Durante la autenticación del cliente y servidor hay un paso que requiere que se cifren los datos con una de las claves de un par de claves asimétricas y que se descifren con la otra clave del par. Se utiliza un resumen de mensaje para proporcionar integridad.

Para obtener una visión general de los pasos implicados en el reconocimiento TLS, consulte “Visión general del reconocimiento SSL/TLS” en la página 19.

Cómo proporciona TLS autenticación

Para la autenticación del servidor, el cliente utiliza la clave pública del servidor para cifrar los datos que ha utilizado para calcular la clave secreta. El servidor puede generar la clave secreta solamente si puede descifrar los datos con la clave privada correcta. La propia serie de bytes aleatorios se cifra con la clave pública del servidor (paso “4” en la página 19 en la visión general).

Para la autenticación del cliente, el servidor utiliza la clave pública del certificado de cliente para descifrar los datos que el cliente envía durante el paso “5” en la página 20 del reconocimiento. El intercambio de mensajes cifrados con las claves secretas que indican que ha finalizado (los pasos “7” en la página 20 y “8” en la página 20 de la visión general) confirma que se ha completado la autenticación.

Si cualquiera de los pasos de autenticación falla, el reconocimiento no se ejecutará correctamente y la sesión finalizará.

El intercambio de certificados digitales durante el reconocimiento TLS forma parte del proceso de autenticación. Para obtener más información acerca de cómo los certificados ofrecen protección contra la suplantación de identidad, consulte la información relacionada. Los certificados necesarios son los siguientes, siendo la CA X la que emite el certificado para el cliente TLS y la CA Y la que emite el certificado para el servidor TLS:

Sólo para la autenticación del servidor, el servidor TLS necesita:

- El certificado personal que la CA Y ha emitido para el servidor
- La clave privada del servidor

y el cliente TLS necesita:

- El certificado de CA de la CA Y

Si el servidor TLS requiere autenticación de cliente, el servidor verifica la identidad del cliente verificando el certificado digital del cliente con la clave pública para la CA que ha emitido el certificado personal al cliente, en este caso CA X. Para la autenticación del servidor y del cliente, el servidor necesita:

- El certificado personal que la CA Y ha emitido para el servidor
- La clave privada del servidor
- El certificado de CA de la CA X

y el cliente necesita:

- El certificado personal que la CA X ha emitido para el cliente
- La clave privada del cliente
- El certificado de CA de la CA Y

Es posible que tanto el servidor como el cliente TLS necesiten otros certificados de CA para formar una cadena de certificados hasta el certificado de CA raíz. Para obtener más información acerca de las cadenas de certificados, consulte la información relacionada.

Qué ocurre durante la verificación de certificados

Como se ha indicado en los pasos “3” en la página 19 y “6” en la página 20 de la visión general, el cliente TLS verifica el certificado del servidor, y el servidor TLS verifica el certificado del cliente. Hay cuatro aspectos en esta verificación:

1. La firma digital se comprueba (consulte “Firmas digitales en SSL/TLS” en la página 23).
2. La cadena de certificados se comprueba; también debe tener certificados de CA intermedios (consulte “Cómo funcionan las cadenas de certificados” en la página 17).
3. Las fechas de activación y caducidad y el período de validez se comprueban.
4. Se comprueba el estado de revocación del certificado (consulte “Trabajar con certificados revocados” en la página 346).

Restablecimiento de claves secretas

Durante el reconocimiento TLS se genera una *clave secreta* que sirve para cifrar los datos que se transfieren del cliente al servidor TLS. La clave secreta utiliza una fórmula matemática que se aplica a los datos para transformar texto plano en texto cifrado ilegible y, texto cifrado en texto plano.

La clave secreta se genera a partir de un texto aleatorio que se envía como parte del reconocimiento y se utiliza para convertir texto plano en texto cifrado. La clave secreta también se utiliza en el algoritmo MAC (Código de autenticación de mensaje), que se utiliza para determinar si se ha modificado un mensaje. Consulte “Resúmenes de mensajes y firmas digitales” en la página 13 para obtener más información.

Si se descubre la clave secreta, podría descifrarse el texto plano de un mensaje a partir del texto cifrado o podría calcularse un resumen del mensaje, que permitiría alterar mensajes sin detectarlo. Incluso para un algoritmo complejo podría llegar a descubrirse el texto plano aplicando cada una de las transformaciones matemáticas posibles al texto cifrado. Para reducir la cantidad de datos que puede descifrarse o modificarse si se descubre la clave secreta, la clave puede negociarse de nuevo periódicamente. Cuando se ha negociado la clave secreta, la clave secreta anterior ya no se podrá utilizar para descifrar datos cifrados con la nueva clave secreta.

Cómo proporciona TLS confidencialidad

TLS utiliza una combinación de cifrado simétrico y asimétrico para garantizar la privacidad de mensajes. Durante el reconocimiento TLS, el cliente y el servidor TLS acuerdan el uso de un algoritmo de cifrado y de una clave secreta compartida que se emplearán sólo para una sesión. Todos los mensajes transmitidos entre el cliente y el servidor TLS se cifran utilizando este algoritmo y esta clave, lo que garantiza la confidencialidad del mensaje incluso si resulta interceptado. Dado que TLS utiliza cifrado asimétrico durante el transporte de la clave secreta compartida, no hay ningún problema de distribución de claves. Para obtener más información acerca de las técnicas de cifrado, consulte “Criptografía” en la página 11.

Cómo proporciona TLS integridad

TLS proporciona integridad de datos calculando un resumen del mensaje. Para obtener más información, consulte “Integridad de datos de mensajes” en la página 485.

El uso de TLS garantiza la integridad de los datos, siempre que la CipherSpec en la definición de canal utilice un algoritmo de hash tal como se describe en la tabla en “Habilitación de CipherSpecs” en la página 428.

En concreto, si la integridad de los datos es una preocupación, debe evitar elegir un CipherSpec cuyo algoritmo hash se muestre como "None". El uso de MD5 también está muy desaconsejado, ya que ahora es muy antiguo y ya no es seguro para la mayoría de los objetivos prácticos.

CipherSpecs y CipherSuites

Los protocolos de seguridad criptográficos deben estar de acuerdo con los algoritmos utilizados por una conexión segura. CipherSpecs y CipherSuites definen combinaciones específicas de algoritmos.

Una CipherSpec identifica una combinación de algoritmo de cifrado y algoritmo MAC (Message Authentication Code). Ambos extremos de una conexión TLS deben estar de acuerdo en la misma CipherSpec para poder comunicarse.

IBM MQ da soporte a los protocolos TLS1.3 y TLS1.2 y CipherSpecs. Sin embargo, puede habilitar las CipherSpecs en desuso, si tiene que hacerlo.

Consulte “Habilitación de CipherSpecs” en la página 428 si desea información sobre:

- CipherSpecs soportadas por IBM MQ
- Cómo se habilitan las CipherSpecs SSL 3.0 y TLS 1.0 en desuso

Importante: Cuando se trata con canales IBM MQ, se debe utilizar una CipherSpec. Cuando se trata con canales Java, canales JMS o canales MQTT debe especificar una CipherSuite.

Para obtener más información sobre CipherSpecs, consulte “Habilitación de CipherSpecs” en la página 428.

Una CipherSuite es una suite de algoritmos de cifrado que utiliza una conexión TLS. Una suite consta de tres algoritmos diferentes:

- El algoritmo de intercambio y autenticación de claves, que se utiliza durante el reconocimiento SSL
- El algoritmo de cifrado, que se utiliza para cifrar los datos
- El algoritmo MAC (Código de autenticación de mensaje), que se utiliza para generar el resumen de mensaje

Hay varias opciones para cada componente de la suite, pero solo ciertas combinaciones son válidas cuando se especifican para una conexión TLS. El nombre de una CipherSuite válida define la combinación de algoritmos utilizados. Por ejemplo, la CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA especifica:

- El algoritmo de intercambio y autenticación de claves RSA
- El algoritmo de cifrado AES, utilizando una clave de 128 bits y una modalidad de encadenamiento de bloques de cifrado (CBC)
- El código de autenticación de mensaje SHA-1 (MAC)

Firmas digitales en SSL/TLS

Una firma digital se crea cifrando una representación de un mensaje. El cifrado utiliza la clave privada del que firma y, por motivos prácticos, suele operar en un resumen del mensaje en lugar de hacerlo en el mensaje propiamente dicho.

Las firmas digitales varían según los datos que se firman, a diferencia de las firmas manuales, que no dependen del contenido del documento que se firma. Si la misma entidad firma digitalmente dos mensajes diferentes, las dos firmas serán diferentes pero ambas pueden verificarse con la misma clave pública, es decir, la clave pública de la entidad que ha firmado los mensajes.

Los pasos del proceso de firma digital son los siguientes:

1. El emisor calcula un resumen de un mensaje y, a continuación, cifra el resumen utilizando la clave privada del emisor, para formar la firma digital.
2. El emisor transmite la firma digital con el mensaje.
3. El receptor descifra la firma digital utilizando la clave pública del emisor y vuelve a generar el resumen del mensaje del emisor.
4. El receptor calcula un resumen del mensaje a partir de los datos del mensaje que recibe y comprueba que los dos resúmenes sean iguales.

La Figura 6 en la página 23 ilustra este proceso.

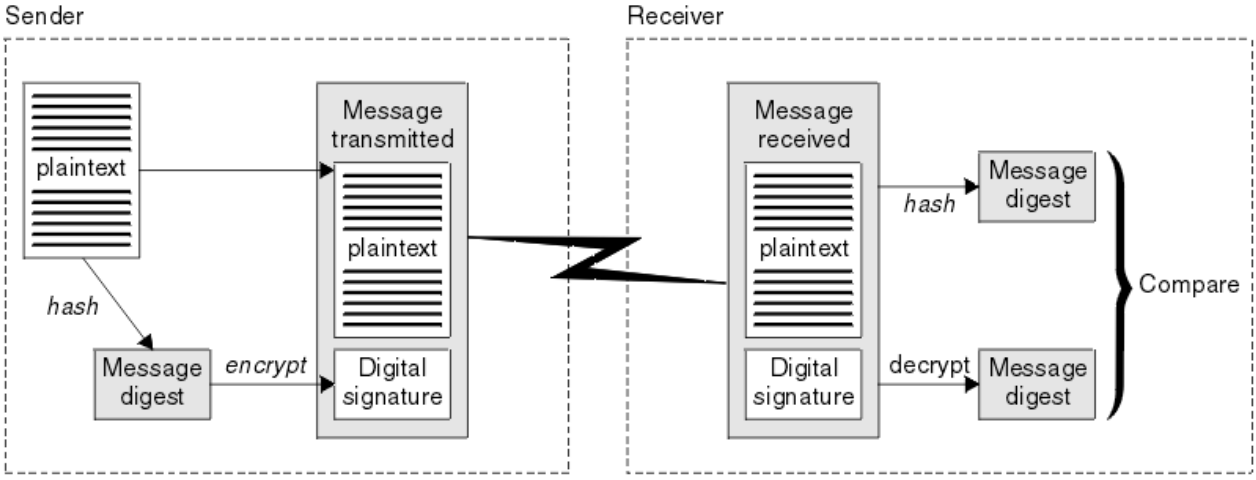


Figura 6. El proceso de firma digital

Si se verifican las dos firmas digitales, el receptor sabe que:

- El mensaje no se ha modificado durante la transmisión.
- El mensaje lo ha enviado la entidad que asegura haberlo enviado.

Las firmas digitales forman parte de los servicios de integridad y autenticación. Las firmas digitales también proporcionan una prueba de origen. Solamente el emisor conoce la clave privada, que proporciona una prueba irrefutable de que el emisor es quien ha originado el mensaje.

Nota: También puede descifrar el mensaje propiamente dicho, lo cual protege la confidencialidad de la información que contiene el mensaje.

Federal Information Processing Standards

El gobierno de EE.UU. ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. El NIST (National Institute for Standards and Technology) es un importante ente al que le preocupan las cuestiones relativas a los sistemas de TI y la seguridad. El NIST propone recomendaciones y genera estándares, entre los que se incluye FIPS (Federal Information Processing Standards).

Uno de estos significativos estándares es FIPS 140-2, que requiere el uso de algoritmos de cifrado fuerte. FIPS 140-2 también especifica los requisitos para que algoritmos de hash se puedan utilizar para proteger los paquetes contra su modificación mientras están en tránsito.

Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC) . El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el certificado deIBM Crypto for C (ICC) y tener en cuenta cualquier consejo proporcionado por NIST. Un módulo FIPS 140-3 de sustitución está actualmente en curso y su estado se puede ver buscándolo en los módulos CMVP de NIST en la lista de procesos.

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los módulos CMVP de NIST en la lista de procesos.

IBM MQ proporciona soporte para FIPS 140-2 cuando se ha configurado para a tal efecto.

Con el tiempo, los analistas desarrollan ataques contra los algoritmos de cifrado y de hash existentes. Se adoptan nuevos algoritmos para poder resistir dichos ataques. FIPS 140-2 se actualiza periódicamente para tener en cuenta estos cambios.

Conceptos relacionados

“Cifrado Suite B de la NSA (National Security Agency)” en la página 24
El gobierno de los Estados Unidos de América ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. La NSA (National Security Agency) de Estados Unidos recomienda un conjunto de algoritmos de cifrado interoperables en su estándar Suite B.

Cifrado Suite B de la NSA (National Security Agency)

El gobierno de los Estados Unidos de América ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. La NSA (National Security Agency) de Estados Unidos recomienda un conjunto de algoritmos de cifrado interoperables en su estándar Suite B.

El estándar Suite B especifica una modalidad de funcionamiento en la que sólo se utiliza un conjunto específico de algoritmos de cifrado. El estándar Suite B especifica lo siguiente:

- El algoritmo de cifrado (AES)
- El algoritmo de intercambio de claves (Elliptic Curve Diffie-Hellman, también conocido como ECDH)
- El algoritmo de firma digital (Elliptic Curve Digital Signature Algorithm, también conocido como ECDSA)
- Los algoritmos de hash (SHA-256 o SHA-384)

Además, el estándar IETF RFC 6460 especifica perfiles compatibles con Suite B que definen la configuración de la aplicación y el comportamiento detallados necesarios para cumplir estándar Suite B. Define dos perfiles:

1. Un perfil compatible con Suite B que puede utilizarse con TLS 1.2. Cuando se configure para el funcionamiento compatible con Suite B, sólo se utilizará el conjunto restringido de algoritmos de cifrado enumerados.
2. Un perfil transitorio para su uso con TLS 1.0 o TLS 1.1. Este perfil permite la interoperatividad con servidores que no sean compatibles con Suite B. Cuando se configura para el funcionamiento transitorio de Suite B, pueden utilizarse algoritmos de cifrado y de hash adicionales.

El estándar Suite B es conceptualmente parecido a FIPS 140-2, porque restringe el conjunto de algoritmos de cifrado permitidos para proporcionar un nivel de seguridad garantizado.

En sistemas AIX, Linux, and Windows , IBM MQ, se puede configurar para que se ajuste al perfil TLS 1.2 compatible con Suite B, pero no da soporte al perfil de transición Suite B. Para obtener más información, consulte "NSA Suite B Cryptography en IBM MQ" en la página 43.

Referencia relacionada

"Federal Information Processing Standards" en la página 24
El gobierno de EE.UU. ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. El NIST (National Institute for Standards and Technology) es un importante ente al que le preocupan las cuestiones relativas a los sistemas de TI y la seguridad. El NIST propone recomendaciones y genera estándares, entre los que se incluye FIPS (Federal Information Processing Standards).

Mecanismos de seguridad de IBM MQ

Esta colección de temas describe mecanismos específicos en IBM MQ que implementan los diversos conceptos de seguridad.

Protocolos de seguridad TLS en IBM MQ

IBM MQ da soporte al protocolo TLS (seguridad de la capa de transporte) para proporcionar seguridad a nivel de enlace para los canales de mensajes y los canales MQI.

Los canales de mensajes y los canales MQI pueden utilizar el protocolo TLS para proporcionar seguridad a nivel de enlace. Un MCA emisor es un cliente TLS y un MCA de respuesta es un servidor TLS.

IBM MQ da soporte a las versiones 1.2 y 1.3 del protocolo TLS. Las versiones anteriores de TLS, así como SSL, no están habilitadas de forma predeterminada, pero pueden serlo si es necesario. Puede especificar los algoritmos de cifrado que utiliza el protocolo TLS suministrando una CipherSpec como parte de la definición de canal.

Consulte "Habilitación de CipherSpecs" en la página 428 para obtener una lista de las CipherSpecs soportadas por IBM MQ y "CipherSpecs en desuso" en la página 444 para las que están en desuso.

Puede utilizar los parámetros SECPROT y SSLCIPH para mostrar el protocolo de seguridad y CipherSpec que se utilizan en un canal.

En cada extremo de un canal de mensajes y en el servidor de un canal MQI, el MCA actúa en nombre del gestor de colas al que está conectado. Durante el reconocimiento TLS, el MCA envía el certificado digital del gestor de colas a su MCA asociado en el otro extremo del canal. El código de IBM MQ en el extremo del cliente de un canal MQI actúa en nombre del usuario de la aplicación de cliente IBM MQ. Durante el reconocimiento TLS, el código IBM MQ envía el certificado digital del usuario al MCA en el extremo de servidor del canal MQI.

Los gestores de colas y los usuarios del cliente IBM MQ no necesitan tener certificados digitales personales asociados cuando actúan como clientes TLS, a menos que se especifique SSLCAUTH(REQUIRED) en el extremo del servidor del canal.

Los certificados digitales se almacenan en un *repositorio de claves*. El atributo de gestor de colas **SSLKeyRepository** especifica la ubicación del depósito de claves que contiene el certificado digital del gestor de colas. En un sistema de cliente IBM MQ, la variable de entorno MQSSLKEYR especifica la ubicación del repositorio de claves que contiene el certificado digital del usuario. De forma alternativa, la aplicación cliente IBM MQ puede especificar su ubicación en el campo **KeyRepository** de la estructura de opciones de configuración de TLS, MQSCO, en una llamada MQCONNX. Consulte los temas relacionados para obtener más información sobre los depósitos de claves y cómo especificar su ubicación.

Soporte para TLS

IBM MQ proporciona soporte para TLS 1.2 y TLS 1.3 en todas las plataformas. Para obtener más información acerca del protocolo TLS, consulte la información en los subtemas.

Clientes de Java y de JMS
Estos clientes utilizan JVM para proporcionar el soporte para TLS.

AIX, Linux, and Windows
El soporte de TLS se instala con IBM MQ.

IBM i
El soporte de TLS está integrado en el sistema operativo IBM i.

z/OS
El soporte de TLS está integrado en el sistema operativo z/OS. El soporte de TLS en z/OS se conoce como *TLS del sistema*.

Para obtener información acerca de los requisitos previos del soporte para TLS en IBM MQ, consulte Requisitos del sistema para IBM MQ.

Conceptos relacionados

"Protocolos de seguridad de cifrado: TLS" en la página 18
Los protocolos de cifrado proporcionan conexiones seguras, permitiendo que dos partes se comuniquen con privacidad e integridad de datos. El protocolo TLS (Transport Layer Security) ha evolucionado a partir del protocolo SSL (Secure Sockets Layer). IBM MQ ofrece soporte para TLS.

Repositorio de claves SSL/TLS

Una conexión TLS mutuamente autenticada requiere un repositorio de claves en cada extremo de la conexión. El repositorio de claves incluye certificados digitales y claves privadas.

En esta información se utiliza el término general *depósito de claves* para describir el almacén de certificados digitales y sus claves privadas asociadas. Se hace referencia al repositorio de claves con diferentes nombres en diferentes plataformas y entornos que dan soporte a TLS:

- IBM i En IBM i: *almacén de certificados*
- En Java y JMS: *almacén de claves* y *almacén de confianza*
- ALW En AIX, Linux, and Windows: *archivo de base de datos de claves*
- z/OS En z/OS: *conjunto de claves*

Para obtener más información, consulte "Certificados digitales" en la página 13 y "Conceptos de TLS (Transport Layer Security)" en la página 19.

Una conexión TLS mutuamente autenticada requiere un repositorio de claves en cada extremo de la conexión. El depósito de claves puede contener los siguientes certificados y solicitudes:

- Un número de certificados de CA de diversas autorizaciones de certificación que permiten al gestor de colas o cliente verificar los certificados que recibe de su asociado en el extremo remoto de la conexión. Los certificados individuales pueden estar dentro de una cadena de certificados.
- Uno o más certificados personales recibidos de una entidad emisora de certificados. Debe asociar un certificado personal diferente a cada gestor de colas o IBM MQ MQI client. Los certificados personales son esenciales en un cliente TLS si la autenticación mutua es necesaria. Si no se requiere autenticación mutua, los certificados personales no son necesarios en el cliente. El depósito de claves podría también contener la clave privada correspondiente a cada certificado personal.
- Las solicitudes de certificados que están en espera de ser firmados por un certificado de CA de confianza.

Para obtener más información acerca de cómo proteger su depósito de claves, consulte "Protección de repositorios de claves de IBM MQ" en la página 27.

La ubicación del repositorio de claves depende de la plataforma que esté utilizando:

IBM i **IBM i**
El depósito de claves es un almacén de certificados. El almacén de certificados del sistema predeterminado se encuentra en `/QIBM/UserData/ICSS/Cert/Server/Default` en el sistema de archivos integrado (IFS). IBM MQ almacena la contraseña para el almacén de certificados en un

archivo de ocultación de contraseña. Por ejemplo, el archivo de ocultación para el gestor de colas QM1 es /QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth.

De forma alternativa, puede especificar que el almacén de certificados del sistema IBM i se utilice en su lugar. Para ello, cambie el valor del atributo **SSLKEYR** del gestor de colas a *SYSTEM. Este valor indica que el gestor de colas debe utilizar el almacén de certificados del sistema, y el gestor de colas se registra para su uso como aplicación con el Gestor de certificados digitales (DCM).

El almacén de certificados también contiene la clave privada para el gestor de colas.

AIX, Linux, and Windows sistemas

El depósito de claves es un almacén de base de datos de claves. Por ejemplo, en AIX and Linux, el archivo de base de datos de claves predeterminado para el gestor de colas QM1 es /var/mqm/qmgrs/QM1/ssl/key.kdb. Si IBM MQ está instalado en la ubicación predeterminada, la vía de acceso equivalente en Windows es C:\ProgramData \IBM \MQ\Qmgrs\QM1\ssl\key.kdb.

Para acceder a un archivo de base de datos de claves, IBM MQ debe proporcionarse la contraseña para la base de datos de claves. Esto se puede hacer directamente o a través de un archivo de ocultación de contraseña. Si se utiliza un archivo de ocultación de contraseña, debe estar en el mismo directorio y tener la misma raíz de archivo que la base de datos de claves, y debe terminar con el sufijo .sth, por ejemplo, /var/mqm/qmgrs/QM1/ssl/key.sth.

Nota: Las tarjetas de hardware criptográfico PKCS #11 pueden contener los certificados y las claves que, de lo contrario, se guardan en un archivo de bases de datos de claves. Cuando los certificados y las claves se guardan en tarjetas PKCS #11, IBM MQ continúa necesitando acceso al archivo de base de datos de claves y a un archivo de ocultación de contraseña.

En sistemas AIX, Linux, and Windows , la base de datos de claves también contiene la clave privada para el certificado personal asociado con el gestor de colas o IBM MQ MQI client.

z/OS

Los certificados se guardan en un conjunto de claves en z/OS.

Otros gestores de seguridad externos (ESM) también utilizan conjuntos de claves para almacenar certificados.

Las claves privadas las gestiona RACF.

Protección de repositorios de claves de IBM MQ

El repositorio de claves para IBM MQ es un archivo. Asegúrese de que solamente el usuario designado pueda acceder al archivo del repositorio de claves. Esto impedirá que un intruso o un usuario no autorizado pueda copiar el archivo del repositorio de claves en otro sistema y establezca, de este modo, un ID de usuario idéntico en dicho sistema para usurpar la identidad del usuario designado.

Los permisos de los archivos dependen del valor de umask del usuario y de qué herramienta se utiliza. En Windows, las cuentas de IBM MQ necesitan el permiso BypassTraverseChecking lo que significa que los permisos de las carpetas en la vía de acceso del archivo no tienen ningún efecto.

Compruebe los permisos de archivos de los archivos del repositorio de claves y asegúrese de que los archivos y la carpeta que los contiene no sean legibles por todos, preferiblemente ni siquiera legibles para grupos.

Hacer el almacén de datos de sólo lectura es una buena práctica, en cualquier sistema que utilice, dejando sólo al administrador como autorizado para habilitar operaciones de escritura para realizar el mantenimiento.

En la práctica, debe proteger todos los almacenes, independientemente de la ubicación y de si están protegidos por contraseña o no; proteja los repositorios de claves.

Etiquetas de certificados digitales, descripción de los requisitos

Al establecer TLS para utilizar certificados digitales, puede que tenga que cumplir algunos requisitos específicos para las etiquetas, en función de la plataforma utilizada y el método que utilice para la conexión.

¿Qué es la etiqueta de certificado?

Una etiqueta de certificado es un identificador exclusivo que representa un certificado digital almacenado en un depósito de claves y que proporciona un nombre legible adecuado con el que hace referencia a un certificado en concreto cuando se realizan funciones de gestión de claves. El usuario asigna la etiqueta de certificado cuando añade un certificado a un depósito de claves por primera vez.

La etiqueta de certificado está separada de los campos **Subject Distinguished Name** o **Subject Common Name** del certificado. Tenga en cuenta que **Subject Distinguished Name** y **Subject Common Name** son campos dentro del propio certificado. Se definen cuando se crea el certificado y no pueden cambiarse. No obstante, puede cambiar la etiqueta asociada a un certificado digital.

Sintaxis de la etiqueta de certificado

Una etiqueta de certificado puede contener letras, números y puntuación con las siguientes condiciones:

- Multi La etiqueta de certificado puede contener hasta 64 caracteres.
- z/OS La etiqueta de certificado puede contener hasta 32 caracteres.
- La etiqueta de certificado puede contener espacios.
- Las etiquetas son sensibles a las mayúsculas y minúsculas.
- En sistemas que utilizan EBCDIC katakana, no puede utilizar caracteres en minúsculas.

Los requisitos adicionales para los valores de etiqueta de certificado se especifican en las siguientes secciones.

¿Cómo se utiliza la etiqueta de certificado?

IBM MQ utiliza etiquetas de certificado para localizar un certificado personal que se envía durante el reconocimiento TLS. De esta manera se elimina la ambigüedad cuando hay más de un certificado personal en el depósito de claves.

Puede establecer la etiqueta de certificado en un valor de su elección. Si no establece un valor, se utiliza una etiqueta predeterminado que sigue un convenio de denominación en función de la plataforma que esté utilizando. Para obtener información detallada, consulte las secciones siguientes sobre plataformas concretas.

Notas:

1. No puede establecer la etiqueta de certificado por su cuenta en los sistemas Java o JMS.
2. Los canales autodefinidos creados mediante una salida de definición automática de canal (CHAD) no pueden establecer la etiqueta de certificado, ya que el reconocimiento TLS se produce en el momento en que se crea el canal. Establecer la etiqueta de certificado en una salida CHAD para los canales de entrada no tiene ningún efecto.

En este contexto, un cliente TLS hace referencia al asociado de la conexión que inicia el reconocimiento, que podría ser un cliente IBM MQ o bien otro gestor de colas.

Durante el reconocimiento TLS, el cliente TLS siempre obtiene y valida un certificado digital del servidor. Con la implementación de IBM MQ, el servidor TLS siempre solicita un certificado del cliente y éste siempre proporciona un certificado al servidor si encuentra uno. Si el cliente no puede localizar un certificado personal, el cliente envía una respuesta no `certificate` al servidor.

El servidor TLS siempre valida el certificado de cliente si se envía uno. Si el cliente no envía un certificado, la autenticación falla si el extremo del canal que actúa como servidor TLS se ha definido con el parámetro **SSLCAUTH** establecido en *REQUIRED* o un valor de parámetro **SSLPEER** establecido.

Tenga en cuenta que los canales de entrada (incluidos el receptor, el solicitante, el clúster receptor, el servidor no calificado y los canales de conexión con el servidor) sólo envían el certificado configurado si la versión de IBM MQ del igual remoto da soporte completo a la configuración de etiqueta de certificado y el canal utiliza una CipherSpecTLS.

Un canal de servidor no calificado es uno que no tiene establecido el campo CONNAME.

En todos los otros casos, el parámetro **CERTLABL** del gestor de colas determina el certificado enviado. En concreto, únicamente reciben el certificado configurado mediante el parámetro **CERTLABL** del gestor de colas los siguientes, independientemente del valor de etiqueta específico de canal:

- Clientes Java y JMS que dan soporte a SNI (Server Name Indication), es decir, certificados canal por canal.
- Las versiones de IBM MQ anteriores a IBM MQ 8.0.
- Clientes .NET gestionados

Además, el certificado utilizado por un canal debe ser adecuado para la CipherSpec del canal; consulte "Certificados digitales y compatibilidad de CipherSpec en IBM MQ" en la página 49 para obtener más información.

IBM MQ 8.0 y posteriores da soporte al uso de varios certificados en el mismo gestor de colas, utilizando una etiqueta de certificado por canal, especificada utilizando el atributo **CERTLABL** en la definición de canal. Los canales de entrada al gestor de colas (por ejemplo, conexión con servidor o receptor) se basan en detectar el nombre de canal utilizando la indicación de nombre de servidor (SNI) de TSL, a fin de presentar el certificado correcto del gestor de colas. Para obtener más información sobre cómo utilizar varios certificados en un gestor de colas, consulte "Cómo proporciona IBM MQ la prestación de varios certificados" en la página 30.

Si un canal se conecta al gestor de colas de destino a través de IBM MQ Internet Pass-Thru (MQIPT), y la ruta MQIPT tiene establecidos **SSLServer** y **SSLClient** , hay dos sesiones TLS separadas entre los puntos finales. MQIPT se puede configurar para permitir que el gestor de colas de destino utilice varios certificados estableciendo el SNI en el nombre de canal o pasando a través del SNI recibido en la conexión de entrada a la ruta. Para obtener más información sobre el soporte de múltiples certificados y MQIPT, consulte Soporte de múltiples certificados IBM MQ con MQIPT.

Si desea más información sobre cómo conectarse a un gestor de colas utilizando la autenticación unidireccional, es decir, cuando un cliente TLS no envía un certificado, consulte Conexión de dos gestores de colas utilizando la autenticación unidireccional.

Sistemas Multiplatforms

Multi

En Multiplatforms, el servidor TLS envía un certificado al cliente.

En el caso de gestores de colas y clientes respectivamente, se busca de forma secuencial un valor que no esté vacío en los siguientes orígenes: El primer valor que no esté vacío determina la etiqueta del certificado. La etiqueta del certificado debe existir en el repositorio de claves. Si no se encuentra un certificado coincidente cuyo formato y combinación de mayúsculas y minúsculas coincida con una etiqueta, se produce un error y el reconocimiento TLS fallará.

Gestores de colas

1. Atributo de etiqueta de certificado de canal **CERTLABL**.
2. Atributo de etiqueta de certificado de gestor de colas **CERTLABL**.
3. Un valor predeterminado con el formato: `ibmwebspheremq` al que se añade el nombre del gestor de colas, en minúsculas. Por ejemplo, para el gestor de colas `QM1`, la etiqueta de certificado predeterminada es `ibmwebspheremqqm1`.

Clientes de IBM MQ

1. Atributo de etiqueta de certificado **CERTLABL** en la definición de canal CLNTCONN.
2. Atributo de estructura MQSCO **CertificateLabel**.
3. Variable de entorno **MQCERTLABL**.
4. Atributo del archivo de cliente `.ini` (en su sección SSL) **CertificateLabel**

5. Un valor predeterminado con el formato: `ibmwebspheremq` al que se añade el ID de usuario de la aplicación de cliente que se está ejecutando, en minúsculas. Por ejemplo, para un ID de usuario USER1, la etiqueta de certificado predeterminada es `ibmwebspheremquser1`.

Sistemas z/OS

z/OS

Los clientes de IBM MQ no están soportados en z/OS. Sin embargo, un gestor de colas de z/OS puede actuar con el rol de cliente TLS cuando inicia una conexión o de un servidor TLS cuando acepta una solicitud de conexión. Los requisitos de la etiqueta de certificado para gestores de colas de z/OS se aplican a ambos roles y son distintos de los requisitos en Multiplatforms.

En el caso de gestores de colas y clientes respectivamente, se busca de forma secuencial un valor que no esté vacío en los siguientes orígenes: El primer valor que no esté vacío determina la etiqueta del certificado. La etiqueta del certificado debe existir en el repositorio de claves. Si no se encuentra un certificado coincidente cuyo formato y combinación de mayúsculas y minúsculas coincida con una etiqueta, se produce un error y el reconocimiento TLS fallará.

1. Atributo de etiqueta de certificado de canal, **CERTLABL**.
2. Si se comparte, el atributo de etiqueta de certificado de grupo de compartición de cola, **CERTQSGL**.

   Si no se comparte, el atributo de etiqueta de certificado de gestor de colas **CERTLABL**.
3. Un valor predeterminado con el formato: `ibmWebSphereMQ` al que se añade el nombre del gestor de colas o del grupo de compartición de cola. Tenga en cuenta que esta serie distingue entre mayúsculas y minúsculas y debe escribir tal como se muestra. Por ejemplo, para el gestor de colas `QM1`, la etiqueta de certificado predeterminada es `ibmWebSphereMQQM1`.
4. Si no se encuentra un certificado con el formato en la opción “3” en la página 30, IBM MQ intenta utilizar el certificado marcado como predeterminado en el conjunto de claves.

Para obtener más información acerca de cómo visualizar el repositorio de claves, consulte “Locating the key repository for a queue manager on z/OS” en la página 316.

Clientes de IBM MQ Java y de IBM MQ JMS

Los clientes de IBM MQ Java y de IBM MQ JMS utilizan los recursos de sus proveedores de JSSE (Java Secure Socket Extension) para seleccionar un certificado personal durante el reconocimiento TLS y, por lo tanto, no están sujetos a los requisitos de las etiquetas de certificados.

El comportamiento predeterminado es que el cliente JSSE examine los certificados del depósito de claves y seleccione el primer certificado personal aceptable que encuentre. Sin embargo, este comportamiento es sólo un valor predeterminado y depende de la implementación del proveedor de JSSE.

Además, la aplicación puede, en tiempo de ejecución, personalizar en gran medida la interfaz JSSE a través de la configuración y el acceso directo. Consulte la documentación que proporciona el proveedor JSSE para obtener detalles específicos.

Para la resolución de problemas o para comprender mejor el reconocimiento que realiza la aplicación de cliente IBM MQ Java en combinación con su proveedor JSSE específico, puede habilitar la depuración estableciendo `javax.net.debug=ssl` en el entorno de la JVM.

Puede establecer la variable en la aplicación durante la configuración o especificando `-Djavax.net.debug=ssl` en la línea de mandatos.

Linux *Cómo proporciona IBM MQ la prestación de varios certificados*

Server Name Indicación (SNI) es una extensión del protocolo TLS que permite a un cliente indicar qué servicio necesita. En terminología de IBM MQ esto equivale a un canal.

La extensión SNI la utiliza IBM MQ para permitir que se especifiquen varios certificados en distintos canales utilizando el parámetro CERTLABL en la definición de canal.

La dirección SNI utilizada por IBM MQ se basa en el nombre de canal que se está solicitando, seguido de un sufijo de `.chl.mq.ibm.com`.

Los nombres de canal de IBM MQ se correlacionan para que sean nombres SNI válidos como se indica a continuación:

- Las letras mayúsculas A a Z se convierten en minúsculas
- Los dígitos `0` a `9` se dejan sin cambios
- Todos los demás caracteres, incluidas las letras minúsculas a a `z`, se convierten en su código de caracteres ASCII hexadecimal de dos dígitos (en minúsculas), seguido de un guión.
  - Las letras minúsculas a a `z` se correlacionan con el hexadecimal `61-` a `7a-` respectivamente
  - porcentaje (`%`) se correlaciona con `25-` hexadecimal
  - guión (`-`) se correlaciona con el hexadecimal `2d-`
  - punto (`.`) se correlaciona con hexadecimal `2e-`
  - barra inclinada (`/`) se correlaciona con el hexadecimal `2f-`
  - El subrayado (`_`) se correlaciona con el hexadecimal `5f-`

En plataformas EBCDIC, el nombre de canal se convierte a ASCII antes de que se aplique esta correlación.

Como ejemplo, el nombre de canal `TO.QMGR1` se correlaciona con una dirección SNI de `to2e-qmgr1.chl.mq.ibm.com`.

Por el contrario, el nombre de canal en minúsculas `to.qmgr1` se correlaciona con la dirección SNI de `74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com`.

Nota: En entornos en los que el URL de SNI generado debe ajustarse a las especificaciones de formato de URL, por ejemplo, cuando un cliente se conecta a un gestor de colas que se ejecuta en Red Hat® OpenShift® a través de una ruta de Red Hat OpenShift , el nombre de canal no debe finalizar con una letra minúscula.

La propiedad **OutboundSNI** de la stanza SSL permite seleccionar si la SNI debe establecerse en el nombre de canal de IBM MQ de destino en el sistema remoto al iniciar una conexión TLS, o bien en el nombre de host. Para obtener más información sobre la propiedad **OutboundSNI**, consulte Stanza SSL del archivo qm.ini y Stanza SSL del archivo de configuración del cliente.

Varios certificados requieren que SNI se establezca en el nombre de canal IBM MQ . Si se utiliza un nombre de host, personalizado o no se utiliza SNI para conectarse a un canal IBM MQ con una etiqueta de certificado configurada, la aplicación de conexión se rechaza con un MQRC_SSL_INITIALIZATION_ERROR y se imprime un mensaje AMQ9673 en los registros de errores del gestor de colas remoto.

Si un canal se conecta al gestor de colas de destino a través de IBM MQ Internet Pass-Thru (MQIPT), MQIPT debe estar configurado para establecer SNI en el nombre de canal, o para pasar a través del SNI recibido en la conexión de entrada a la ruta, para permitir que el gestor de colas de destino utilice varios certificados. Para obtener más información sobre el soporte de múltiples certificados y MQIPT, consulte Soporte de múltiples certificados IBM MQ con MQIPT.

Para obtener más información sobre cómo se utiliza esta propiedad, consulte Conexión a un gestor de colas desplegado en un clúster de Red Hat OpenShift.

Renovación del depósito de claves del gestor de colas
Cuando cambia el contenido de un depósito de claves, los procesos existentes del gestor de colas no recogen el nuevo contenido hasta que se emite un mandato REFRESH SECURITY TYPE (SSL) o se reinicia el gestor de colas.

Para obtener más información sobre el mandato REFRESH SECURITY TYPE(SSL), consulte REFRESH SECURITY.

Si el gestor de colas crea un nuevo proceso de canal (utilizando `amqrmppa` o **runmqchl**) después de cambiar el contenido del almacén de claves, el nuevo proceso se inicia utilizando los nuevos certificados inmediatamente, mientras que los procesos existentes continúan utilizando su copia en memoria caché

del almacén de claves. Consulte “Cuándo entran en vigor los cambios en los certificados o en el repositorio de claves en AIX, Linux, and Windows” en la página 312 para obtener más detalles.

Tenga en cuenta que varios canales en ejecución podrían estar utilizando distintas versiones del repositorio de claves hasta que emita un mandato REFRESH SECURITY TYPE (SSL).

También puede renovar un repositorio de claves utilizando mandatos PCF o IBM MQ Explorer. Para obtener más información, consulte el Mandato MQCMD_REFRESH_SECURITY y el tema *Renovación de la seguridad TLS* en la sección de IBM MQ Explorer de esta documentación de producto.

Conceptos relacionados

“Renovación de la vista de un cliente del contenido de repositorio de claves SSL/TLS y valores SSL/TLS” en la página 32
Para actualizar la aplicación cliente con el contenido renovado del repositorio de claves, debe detener y reiniciar la aplicación cliente.

Renovación de la vista de un cliente del contenido de repositorio de claves SSL/TLS y valores SSL/TLS
Para actualizar la aplicación cliente con el contenido renovado del repositorio de claves, debe detener y reiniciar la aplicación cliente.

No se puede renovar la seguridad en un cliente IBM MQ; no hay ningún equivalente al mandato REFRESH SECURITY TYPE(SSL) para clientes (consulte REFRESH SECURITY) para obtener más información.

Para actualizar la aplicación cliente con el contenido renovada del repositorio de claves, debe detener y reiniciar la aplicación, siempre que cambie el certificado de seguridad.

Si reiniciar el canal renueva la configuración, y si la aplicación tiene lógica de reconexión, es posible renovar la seguridad en el cliente emitiendo el mandato STOP CHL STATUS(INACTIVE).

Conceptos relacionados

“Renovación del depósito de claves del gestor de colas” en la página 31
Cuando cambia el contenido de un depósito de claves, los procesos existentes del gestor de colas no recogen el nuevo contenido hasta que se emite un mandato REFRESH SECURITY TYPE (SSL) o se reinicia el gestor de colas.

Protección por contraseña MQCSP

Las credenciales de autenticación que se especifican en la estructura MQCSP se pueden proteger utilizando la característica de protección de contraseña MQCSP de IBM MQ o se pueden cifrar utilizando el cifrado TLS.

Las aplicaciones de IBM MQ client pueden proporcionar un ID de usuario y una contraseña cuando se conectan a un gestor de colas. V 9.4.0 A partir de IBM MQ 9.4.0, las aplicaciones también pueden proporcionar una señal de autenticación como método alternativo de autenticación. Estas credenciales se envían al gestor de colas en una estructura MQCSP.

Si el canal utiliza el cifrado TLS, las credenciales de MQCSP se cifran de acuerdo con la especificación de cifrado TLS. Si el canal no utiliza el cifrado TLS, IBM MQ puede proteger estas credenciales antes de que se envíen a través de la red, para evitar el envío de credenciales a través de una red en texto sin formato. La característica IBM MQ que protege estas credenciales se denomina protección por contraseña MQCSP.

Si se utiliza la protección de contraseña MQCSP, se protegen los datos siguientes en la estructura MQCSP:

- La contraseña, si el campo `MQCSP.AuthenticationType` se establece en `MQCSP_AUTH_USER_ID_AND_PW`.
- V 9.4.0 La señal de autenticación, si el campo `MQCSP.AuthenticationType` se establece en `MQCSP_AUTH_ID_TOKEN`.

Importante: La protección por contraseña MQCSP es útil para fines de prueba y desarrollo porque utilizar la protección por contraseña MQCSP es más sencillo que establecer el cifrado TLS, pero no es tan seguro. Para fines de producción, utilice el cifrado TLS en lugar de la protección de contraseña de IBM MQ , especialmente cuando la red entre el cliente y el gestor de colas no es de confianza, ya que el cifrado TLS es más seguro.

Si le preocupa qué cifrado se está utilizando y cuánta protección ofrece, debe utilizar el cifrado TLS completo. Con TLS, los algoritmos se conocen públicamente y puede seleccionar el adecuado para su empresa utilizando el atributo de canal **SSLCIPH** .

Para obtener más información sobre la estructura MQCSP, consulte Estructura MQCSP.

Las credenciales de la estructura MQCSP se protegen utilizando la protección por contraseña de IBM MQ si se cumplen todas las condiciones siguientes:

- Los dos extremos de la conexión utilizan IBM MQ 8.0 o posterior.
- El canal no está utilizando el cifrado TLS. Un canal no utiliza el cifrado TLS si el canal tiene un atributo **SSLCIPH** en blanco, o si el atributo **SSLCIPH** está establecido en una especificación de cifrado que no proporciona cifrado. Los cifrado nulos, por ejemplo, NULL_SHA, no proporcionan cifrado.
- El campo `MQCSP.AuthenticationType` se establece en `MQCSP_AUTH_USER_ID_AND_PWD` o `MQCSP_AUTH_ID_TOKEN`. Para obtener más información sobre el campo `MQCSP.AuthenticationType` , consulte **AuthenticationType**.
- Si el cliente es IBM MQ Explorer y la modalidad de compatibilidad de identificación de usuario no está habilitada. Esta modalidad no es la modalidad predeterminada que utiliza IBM MQ Explorer para enviar un ID de usuario y una contraseña. Esta condición sólo es aplicable a IBM MQ Explorer.

Si no se cumple alguna de estas condiciones, las credenciales no se protegen con la protección de contraseña MQCSP. Si el valor del atributo **PasswordProtection** prohíbe que las credenciales se envíen en texto sin formato, y el canal no utiliza el cifrado TLS, la conexión falla y se devuelve un código de razón MQRC_PASSWORD_PROTECTION_ERROR (2594).

El valor de configuración `PasswordProtection`

El atributo **PasswordProtection** de la stanza **Channels** de los archivos de configuración del cliente y del gestor de colas puede impedir que las credenciales se envíen en texto sin formato.

Nota: Este atributo sólo es relevante para las conexiones que no utilizan el cifrado TLS. Las credenciales se cifran utilizando TLS en lugar de protegerse con la protección de contraseña MQCSP si la conexión utiliza el cifrado TLS.

El atributo se puede establecer en uno de los valores siguientes. El valor predeterminado es `compatible`.

compatible
Las credenciales se envían en texto sin formato si el gestor de colas o el cliente está ejecutando una versión de IBM MQ anterior a IBM MQ 8.0. Es decir, las credenciales se pueden enviar a través de una red en texto sin formato para mantener la compatibilidad con las versiones de IBM MQ que no dan soporte a la protección por contraseña MQCSP.

Las credenciales están protegidas por la protección de contraseña MQCSP si tanto el gestor de colas como el cliente ejecutan una versión de IBM MQ en IBM MQ 8.0 o posterior.

La conexión falla antes de que se envíen las credenciales si tanto el gestor de colas como el cliente están ejecutando una versión de IBM MQ en IBM MQ 8.0 o posterior, y el campo `MQCSP.AuthenticationType` no está establecido en `MQCSP_AUTH_USER_ID_AND_PW` o `MQCSP_AUTH_ID_TOKEN`.

always
Las credenciales no se deben enviar a través de una red no protegida.

Las credenciales están protegidas por la protección de contraseña MQCSP si tanto el gestor de colas como el cliente ejecutan una versión de IBM MQ en IBM MQ 8.0 o posterior.

La conexión falla antes de que se envíen las credenciales en los casos siguientes:

- El campo `MQCSP.AuthenticationType` no está establecido en `MQCSP_AUTH_USER_ID_AND_PW` o `MQCSP_AUTH_ID_TOKEN`.
- El gestor de colas o el cliente está ejecutando una versión de IBM MQ anterior a IBM MQ 8.0.

opcional
Las credenciales se protegen mediante la protección de contraseña MQCSP si tanto el gestor de colas como el cliente ejecutan una versión de IBM MQ en IBM MQ 8.0 o posterior, y el campo `MQCSP.AuthenticationType` se establece en `MQCSP_AUTH_USER_ID_AND_PW` o `MQCSP_AUTH_ID_TOKEN`. De lo contrario, las credenciales se envían en texto sin formato.

aviso
Cualquier cliente puede enviar credenciales de texto sin formato. Si se reciben credenciales de texto sin formato, el mensaje de aviso AMQ9297W se graba en los registros de errores del gestor de colas.

Esta opción sólo se puede especificar en el archivo de configuración del gestor de colas.

Para los clientes Java y JMS , el comportamiento del atributo **PasswordProtection** cambia en función de si el cliente utiliza la modalidad de compatibilidad o la modalidad MQCSP:

- Si los clientes Java y JMS están funcionando en modalidad de compatibilidad, no se utiliza una estructura MQCSP para enviar el ID de usuario y la contraseña cuando el cliente se conecta. Por lo tanto, el comportamiento del atributo **PasswordProtection** es el mismo que el comportamiento descrito para los clientes que ejecutan una versión de IBM MQ anterior a IBM MQ 8.0.
- Si los clientes Java y JMS están operando en modalidad MQCSP, el comportamiento del atributo **PasswordProtection** es el comportamiento descrito.

Para obtener más información sobre la autenticación de conexión con clientes Java y JMS , consulte “Autenticación de conexión con el cliente Java” en la página 86.

Protección de contraseña MQCSP y MQIPT

V 9.4.0

Si un cliente se conecta a un gestor de colas a través de IBM MQ Internet Pass-Thru (MQIPT), la ruta MQIPT se puede configurar para añadir o eliminar el cifrado TLS. Es decir, la ruta de MQIPT se puede configurar con `SSLServer=true` y `SSLClient=false`, o `SSLServer=true` y `SSLClient=false`. En esta situación, es posible que el cliente y el gestor de colas no concuerden con un algoritmo de protección de contraseña, ya que un extremo del canal está utilizando el cifrado TLS y el otro no. Esto hace que la conexión falle con el código de razón MQRC_PASSWORD_PROTECTION_ERROR (2594).

A partir de IBM MQ 9.4.0, MQIPT puede añadir o eliminar protección para credenciales en estructuras MQCSP, para mantener la compatibilidad entre el cliente y el gestor de colas para rutas MQIPT que añaden o eliminan cifrado TLS. La protección de contraseña MQCSP en MQIPT se configura utilizando la propiedad de ruta **PasswordProtection** .

El valor predeterminado de la propiedad **PasswordProtection** es `required`. Este valor significa que MQIPT puede añadir, pero no eliminar, la protección de contraseña MQCSP. Las conexiones a una ruta de MQIPT que añade cifrado TLS pueden fallar con el código de razón MQRC_PASSWORD_PROTECTION_ERROR (2594) con este valor de **PasswordProtection**. Para resolver este problema, establezca el valor de la propiedad **PasswordProtection** en `compatible` en la configuración de ruta de MQIPT .

Para obtener más información sobre la propiedad **PasswordProtection** en MQIPT, consulte PasswordProtection.

gestor de certificados digitales (DCM)

Utilice DCM para gestionar certificados digitales y claves privadas en IBM i.

El Gestor de certificados digitales (DCM) le permite gestionar certificados digitales y utilizarlos en aplicaciones seguras en el servidor IBM i. Con el Gestor de certificados digitales, puede solicitar y procesar certificados digitales de Entidades emisoras de certificados (CA) o de terceros. También puede actuar como una entidad emisora de certificados local para crear y gestionar certificados digitales para sus usuarios.

DCM también da soporte al uso de Listas de revocación de certificados (CRL) para proporcionar un proceso de validación de certificados y aplicaciones más potente. Puede utilizar DCM para definir la

ubicación donde reside una CRL de una entidad emisora de certificados específica en un servidor LDAP para que IBM MQ pueda verificar que no se ha revocado un certificado específico.

DCM da soporte y puede detectar automáticamente certificados en diversos formatos. Cuando DCM detecta un certificado codificado PKCS #12 o un certificado PKCS #7 que contiene datos cifrados, solicita automáticamente al usuario que escriba la contraseña que se ha utilizado para cifrar el certificado. DCM no solicita certificados PKCS #7 que no contengan datos cifrados.

DCM proporciona una interfaz de usuario basada en navegador que se puede utilizar para gestionar certificados digitales para las aplicaciones y los usuarios. La interfaz de usuario está dividida en dos secciones principales: una sección de navegación y una sección de tareas.

Utilice la sección de navegación para seleccionar las tareas para gestionar certificados o las aplicaciones que los utilizan. Algunas tareas individuales se muestran directamente en la sección de navegación principal, pero la mayoría de las tareas de la sección de navegación se organizan en categorías. Por ejemplo, Gestionar certificados es una categoría de tareas que contiene diversas tareas guiadas individuales, como por ejemplo Ver certificado, Renovar certificado e Importar certificado. Si un elemento de la sección de navegación es una categoría que contiene más de una tarea, se muestra una flecha a la izquierda del mismo. La flecha indica que cuando se selecciona el enlace de la categoría, se visualiza una lista ampliada de tareas, que le permite elegir qué tarea desea realizar.

Para obtener información importante sobre DCM, consulte las siguientes publicaciones IBM Redbooks:

- *IBM i Seguridad de red con conexión: OS/400 V5R1 Mejoras DCM y criptográficas*, SG24-6168. Concretamente, consulte los apéndices para obtener información esencial sobre la configuración del sistema IBM i como CA local.
- *AS/400 Internet Security: Developing a Digital Certificate Infrastructure*, SG24-5659. En concreto, véase el capítulo 5. *Digital Certificate Manager para AS/400* , que explica el AS/400 DCM.

Estándares federales de procesamiento de la información (FIPS)

En este tema se presenta los estándares federales de procesamiento de la información (FIPS) Cryptomodule Validation Program del US National Institute of Standards and Technology y las funciones de cifrado que se pueden utilizar en canales TLS.

Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC) . El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el certificado deIBM Crypto for C (ICC) y tener en cuenta cualquier consejo proporcionado por NIST. Un módulo FIPS 140-3 de sustitución está actualmente en curso y su estado se puede ver buscándolo en los módulos CMVP de NIST en la lista de procesos.

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los módulos CMVP de NIST en la lista de procesos.

Esta información se aplica a las siguientes plataformas:

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

ALW Para obtener más información sobre la conformidad con FIPS 140-2 de una conexión TLS de IBM MQ en AIX, Linux, and Windows, consulte “Federal Information Processing Standards (FIPS) para AIX, Linux, and Windows” en la página 36.

z/OS Para obtener más información sobre la conformidad con FIPS 140-2 de una conexión TLS de IBM MQ en z/OS, consulte “Federal Information Processing Standards (FIPS) for z/OS” en la página 39.

Si el hardware de cifrado está presente, los módulos de cifrado utilizados por IBM MQ se pueden configurar de modo que sean los proporcionados por el fabricante del hardware. En este caso, la configuración sólo será compatible con FIPS si dichos módulos de cifrado tienen certificación FIPS.

Con el tiempo, los Estándares federales de procesamiento de la información (FIPS) se actualizan para reflejar nuevos estándares frente a algoritmos y protocolos de cifrado. Por ejemplo, algunas CipherSpecs pueden dejar de certificarse con FIPS. Cuando se producen estos cambios, IBM MQ también se actualiza para implementar el último estándar. Como resultado, es posible que vea cambios en el comportamiento después de aplicar el mantenimiento.

Conceptos relacionados

"Especificación de que sólo se utilizan CipherSpecs certificadas por FIPS en el tiempo de ejecución del cliente MQI" en la página 274
Cree sus propios repositorios de claves utilizando software compatible con FIPS y, a continuación, especifique que el canal debe utilizar las CipherSpecs certificadas por FIPS.

Tareas relacionadas

Habilitación de TLS en IBM MQ classes for Java

Utilización de TLS (seguridad de la capa de transporte) con IBM MQ classes for JMS

Referencia relacionada

Propiedades TLS de los objetos JMS

"Mandatos runmqakm y runmqktool en AIX, Linux, and Windows" en la página 551
En sistemas AIX, Linux, and Windows , utilice los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool) para gestionar claves y certificados.

"Federal Information Processing Standards" en la página 24
El gobierno de EE.UU. ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. El NIST (National Institute for Standards and Technology) es un importante ente al que le preocupan las cuestiones relativas a los sistemas de TI y la seguridad. El NIST propone recomendaciones y genera estándares, entre los que se incluye FIPS (Federal Information Processing Standards).

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

Cuando la criptografía es necesaria en un canal SSL/TLS en sistemas AIX, Linux, and Windows , IBM MQ utiliza un paquete de criptografía denominado IBM Crypto for C (ICC). En las plataformas AIX, Linux, and Windows , el software ICC ha pasado el Programa de validación de criptomódulos FIPS (Federal Information Processing Standards) del Instituto Nacional de Estándares y Tecnología de Estados Unidos, en el nivel 140-2.

Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC) . El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el certificado deIBM Crypto for C (ICC) y tener en cuenta cualquier consejo proporcionado por NIST. Un módulo FIPS 140-3 de sustitución está actualmente en curso y su estado se puede ver buscándolo en los módulos CMVP de NIST en la lista de procesos.

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los módulos CMVP de NIST en la lista de procesos.

La conformidad con FIPS 140-2 de una conexión TLS de IBM MQ en sistemas AIX, Linux, and Windows es la siguiente:

- Para todos los canales de mensajes de IBM MQ message channels (excepto los tipos de canal CLNTCONN), la conexión es compatible con FIPS si se cumplen las condiciones siguientes:
  - La versión instalada de IBM Global Security Kit (GSKit) ICC ha sido certificada compatible con FIPS 140-2 en la versión instalada del sistema operativo y la arquitectura de hardware.
  - El atributo SSLFIPS del gestor de colas se ha establecido en YES.
  - Todos los repositorios de claves se han creado y manipulado utilizando únicamente software compatible con FIPS como por ejemplo, **runmqakm** con la opción `-fips`.
  - El acceso a todos los repositorios de claves se proporciona utilizando un archivo de ocultación y no el atributo **KEYRPWD** del gestor de colas.

- Para todas las aplicaciones IBM MQ MQI client , la conexión utiliza GSKit y es compatible con FIPS si se cumplen las condiciones siguientes:
  - La versión instalada de GSKit ICC ha sido certificada compatible con FIPS 140-2 en la versión instalada del sistema operativo y la arquitectura de hardware.
  - Ha especificado que sólo se utilizará el cifrado certificado por FIPS, tal como se describe en el tema relacionado para el cliente MQI.
  - Todos los repositorios de claves se han creado y manipulado utilizando únicamente software compatible con FIPS como por ejemplo, **runmqakm** con la opción `-fips`.
  - El acceso a todos los repositorios de claves se proporciona utilizando un archivo de ocultación y no el mecanismo de contraseña del repositorio de claves.
- Para las aplicaciones de IBM MQ classes for Java que utilizan la modalidad de cliente, la conexión utiliza las implementaciones TLS de JRE y es compatible con FIPS si se cumplen las condiciones siguientes:
  - JRE (Java Runtime Environment) que se utiliza para ejecutar la aplicación cumple con la norma FIPS en la versión de sistema operativo y la arquitectura de hardware instalada.
  - Ha especificado que sólo se utilizará el cifrado certificado por FIPS, tal como se describe en el tema relacionado para el cliente Java.
  - Todos los repositorios de claves se han creado y manipulado utilizando únicamente software compatible con FIPS como por ejemplo, **runmqakm** con la opción `-fips`.
- Para las aplicaciones de IBM MQ classes for JMS que utilizan la modalidad de cliente, la conexión utiliza las implementaciones TLS de JRE y es compatible con FIPS si se cumplen las condiciones siguientes:
  - JRE (Java Runtime Environment) que se utiliza para ejecutar la aplicación cumple con la norma FIPS en la versión de sistema operativo y la arquitectura de hardware instalada.
  - Ha especificado que sólo se utilizará el cifrado certificado por FIPS, tal como se describe en el tema relacionado para el cliente JMS.
  - Todos los repositorios de claves se han creado y manipulado utilizando únicamente software compatible con FIPS como por ejemplo, **runmqakm** con la opción `-fips`.
- Para aplicaciones cliente .NET no gestionadas, la conexión utiliza GSKit y es compatible con FIPS si se cumplen las condiciones siguientes:
  - La versión instalada de GSKit ICC ha sido certificada compatible con FIPS 140-2 en la versión instalada del sistema operativo y la arquitectura de hardware.
  - Ha especificado que sólo se utilizará el cifrado certificado por FIPS, tal como se describe en el tema relacionado para el cliente .NET.
  - Todos los repositorios de claves se han creado y manipulado utilizando únicamente software compatible con FIPS como por ejemplo, **runmqakm** con la opción `-fips`.
  - El acceso a todos los repositorios de claves se proporciona utilizando un archivo de ocultación y no el mecanismo de contraseña del repositorio de claves.
- Para aplicaciones cliente XMS .NET no gestionadas, la conexión utiliza GSKit y es compatible con FIPS si se cumplen las condiciones siguientes:
  - La versión instalada de GSKit ICC ha sido certificada compatible con FIPS 140-2 en la versión instalada del sistema operativo y la arquitectura de hardware.
  - Ha especificado que sólo se va a utilizar la criptografía certificada por FIPS, tal como se describe en la documentación de XMS .NET .
  - Todos los repositorios de claves se han creado y manipulado utilizando únicamente software compatible con FIPS como por ejemplo, **runmqakm** con la opción `-fips`.
  - El acceso a todos los repositorios de claves se proporciona utilizando un archivo de ocultación y no el mecanismo de contraseña del repositorio de claves.

Todas las plataformas soportadas cuentan con el certificado FIPS 140-2, excepto cuando se indique en el archivo Readme que se incluye con cada fixpack o paquete de actualización.

Para las conexiones TLS que utilizan GSKit, el componente que está certificado FIPS 140-2 se denomina *ICC*. Es la versión de este componente la que determina la conformidad con FIPS de GSKit en cualquier plataforma determinada. Para determinar la versión de ICC instalada actualmente, ejecute el mandato **dspmqver -p 64 -v** .

A continuación se muestra un extracto de ejemplo de la salida **dspmqver -p 64 -v** relacionada con 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.
@ (#) Reservados todos los derechos. 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:
```

La declaración de certificación NIST para GSKit ICC 8 (incluido en GSKit 8) se puede encontrar en la siguiente dirección: Programa de validación de módulo criptográfico.

Si el hardware de cifrado está presente, los módulos de cifrado utilizados por IBM MQ se pueden configurar de modo que sean los proporcionados por el fabricante del hardware. En este caso, la configuración sólo será compatible con FIPS si dichos módulos de cifrado tienen certificación FIPS.

Restricciones de Triple DES aplicadas al operar en conformidad con FIPS 140-2

Cuando IBM MQ se ha configurado para que funcione en conformidad con FIPS 140-2, se aplican restricciones adicionales en relación con las CipherSpecs de Triple DES (3DES). Estas restricciones permiten la conformidad con la recomendación NIST SP800-67 de los Estados Unidos.

1. Todas las partes de la clave Triple DES deben ser exclusivas.
2. Ninguna parte de la clave Triple DES puede ser una clave débil, semi-débil o posiblemente débil de acuerdo con las definiciones de NIST SP800-67.
3. No pueden transmitirse más de 32 GB de datos por medio de la conexión antes de que se tenga que producir un restablecimiento de clave secreta. De forma predeterminada, IBM MQ no restablece la clave de sesión secreta, por lo que este restablecimiento se debe configurar. Si no habilita el restablecimiento de la clave secreta cuando se utiliza una CipherSpec Triple DES y la conformidad con FIPS 140-2 da como resultado el cierre de la conexión con el error AMQ9288 después de superar el número de bytes máximo. Para obtener información sobre cómo configurar el restablecimiento de la clave secreta, consulte “Restablecimiento de claves secretas SSL y TLS” en la página 475.

IBM MQ genera claves de sesión DES triple que ya cumplen con las reglas 1 y 2. Sin embargo, para satisfacer la tercera restricción, debe habilitar el restablecimiento de clave secreta cuando utilice Triple DES CipherSpecs en una configuración de FIPS 140-2. Como alternativa, puede evitar el uso de Triple DES.

Conceptos relacionados

“Especificación de que sólo se utilizan CipherSpecs certificadas por FIPS en el tiempo de ejecución del cliente MQI” en la página 274
Cree sus propios repositorios de claves utilizando software compatible con FIPS y, a continuación, especifique que el canal debe utilizar las CipherSpecs certificadas por FIPS.

Tareas relacionadas

Habilitación de TLS en IBM MQ classes for Java

Utilización de TLS (seguridad de la capa de transporte) con IBM MQ classes for JMS

Referencia relacionada

Propiedades TLS de los objetos JMS

"Mandatos runmqakm y runmqktool en AIX, Linux, and Windows" en la página 551
En sistemas AIX, Linux, and Windows , utilice los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool) para gestionar claves y certificados.

"Federal Information Processing Standards" en la página 24
El gobierno de EE.UU. ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. El NIST (National Institute for Standards and Technology) es un importante ente al que le preocupan las cuestiones relativas a los sistemas de TI y la seguridad. El NIST propone recomendaciones y genera estándares, entre los que se incluye FIPS (Federal Information Processing Standards).

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 24
El gobierno de EE.UU. ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. El NIST (National Institute for Standards and Technology) es un importante ente al que le preocupan las cuestiones relativas a los sistemas de TI y la seguridad. El NIST propone recomendaciones y genera estándares, entre los que se incluye FIPS (Federal Information Processing Standards).

Multi

Verificación de la configuración de TLS del gestor de colas con `mqcertck`

El mandato **MQCERTCK** es una herramienta para buscar errores comunes en la configuración de TLS del gestor de colas, y proporciona algunas sugerencias para resolver problemas.

Introducción

El mandato **`mqcertck`** comprueba lo siguiente:

- La existencia y los permisos del repositorio de claves del gestor de colas, al que se hace referencia en el atributo **SSLKEYR** del gestor de colas.
- La existencia y la validez del certificado del gestor de colas, al que se hace referencia en el atributo **CERTLABL** del gestor de colas.
- La existencia y la validez de los certificados a los que se hace referencia en los atributos **CERTLABL** del canal habilitado para TLS.
- El repositorio de claves y los certificados de las aplicaciones cliente, que incluye comprobar si los certificados están autorizados con el gestor de colas.

Nota: El mandato **`mqcertck`** no está disponible en z/OS o IBM i.

Utilización

Para utilizar el mandato **`mqcertck`**, ejecute el mandato `mqcertck`, junto con los parámetros necesarios, y los parámetros opcionales que necesite, desde una línea de mandatos.

Consulte mqcertck para obtener una descripción del mandato y de los parámetros que acepta el mandato.

Ejemplo

Ha terminado de configurar el gestor de colas `QM1` para permitir las conexiones TLS de los clientes que se conectan al canal SVRCONN del gestor de colas.

Utiliza la característica de múltiples certificados y, por lo tanto, tanto el gestor de colas como el canal tienen un etiqueta de certificado especificada en sus atributos **CERTLABL**. Al crear el canal, ha cometido un error en el atributo **CERTLABL** del canal, por lo que cuando un cliente intenta conectarse, el gestor de colas devuelve un código de retorno 2393 de MQRC_SSL_INITIALIZATION_ERROR.

Antes de activar el gestor de colas, debe utilizar el mandato **`mqcertck`** para verificar la configuración de TLS del gestor de colas.

Puede ejecutar el mandato `mqcertck QM1` y recibir la salida siguiente:

```
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.
|
+------------------------------------------------------------
```

Esta salida le solicita que compruebe la definición de canal para el canal de conexión de servidor MQCERTCK.CHANNEL. Aquí verá el error que ha cometido y puede corregirlo antes de volver a ejecutar el mandato `mqcertck` para verificar que se ha resuelto el problema.

Verificación de las conexiones de cliente

El mandato **mqcertck** tiene la capacidad de verificar los repositorios de claves de cliente, así como la configuración TLS del gestor de colas. Para ello, **mqcertck** debe poder acceder al repositorio de claves del cliente desde la máquina que ejecuta el gestor de colas.

Al ejecutar el mandato **mqcertck** , si proporciona el parámetro **-clientkeyr** con la ubicación del repositorio de claves de cliente (excluyendo la extensión), **mqcertck** comprueba este repositorio de claves en el gestor de colas.

Si sabe qué canal utilizará el cliente para conectarse al gestor de colas, puede especificarlo con el distintivo **-clientchannel** .

Si el cliente está utilizando la autenticación mutua para conectarse al gestor de colas, puede utilizar el parámetro **-clientusername** o **-clientlabel** , para indicar al mandato **mqcertck** qué certificado debe utilizar en el repositorio de claves del cliente.

Si utiliza el certificado predeterminado y no proporciona una etiqueta de certificado a la aplicación cliente, puede utilizar los parámetros **-clientusername** y **username** que ejecutan esta aplicación.

Durante la operación del mandato **mqcertck** , el mandato genera la etiqueta de certificado `ibmwebspheremqXXXX` donde XXXX es el valor pasado en el parámetro **-clientusername** .

Para verificar completamente el repositorio de claves del cliente, el mandato **mqcertck** crea una conexión ficticia utilizando IBM Global Security Kit (GSKit). Para ello, el mandato debe tener un puerto disponible al que pueda enlazarse durante las pruebas de cliente. El puerto predeterminado utilizado es 5857, pero si este ya está en uso puede especificar un puerto distinto que se utilizará durante las pruebas de cliente.

Nota: Aunque el mandato **mqcertck** se enlaza con un puerto, no se utiliza ninguna comunicación externa mediante **mqcertck** y todas las pruebas se realizan localmente.

SSL/TLS en el IBM MQ MQI client

IBM MQ ofrece soporte para TLS en los clientes. Puede adaptar el uso de TLS de varias maneras.

IBM MQ proporciona soporte TLS para IBM MQ MQI clients en sistemas AIX, Linux, and Windows . Si está utilizando IBM MQ classes for Java, consulte Utilización de IBM MQ classes for Java y si está utilizando IBM MQ classes for JMS, consulte Utilización de IBM MQ classes for JMS. El resto de esta sección no es aplicable a los entornos de Java o JMS.

Puede especificar el repositorio de claves para un IBM MQ MQI client con el valor MQSSLKEYR en el archivo de configuración de cliente IBM MQ, o cuando su aplicación realice una llamada MQCONNX. Dispone de tres opciones para especificar que un canal utiliza TLS:

- Utilizar una tabla de definiciones de canal
- Utilizar la estructura de opciones de configuración de SSL, MQSCO, en una llamada MQCONNX
- Utilizar Active Directory (en sistemas Windows)

No puede utilizar la variable de entorno MQSERVER para especificar que un canal utiliza TLS.

Puede seguir ejecutando las aplicaciones IBM MQ MQI client existentes sin TLS, siempre y cuando no se especifique TLS en el otro extremo del canal.

Si se efectúan cambios en una máquina cliente en el contenido del repositorio de claves TLS, la ubicación del repositorio de claves TLS, la información de autenticación o los parámetros de hardware de cifrado, debe finalizar todas las conexiones TLS para reflejar estos cambios en los canales de conexión de cliente que la aplicación utiliza para conectarse al gestor de colas. Una vez que hayan finalizado todas las conexiones, reinicie los canales TLS. Se utilizarán los nuevos valores TLS. Estos valores son análogos a los actualizados por el mandato REFRESH SECURITY TYPE(SSL) en los sistemas del gestor de colas.

Cuando el IBM MQ MQI client se ejecuta en un sistema AIX, Linux, and Windows con hardware criptográfico, debe configurar dicho hardware con la variable de entorno MQSSLCRYP. Esta variable es equivalente al parámetro SSLCRYP del mandato MQSC ALTER QMGR. Consulte ALTER QMGR para obtener una descripción del parámetro SSLCRYP del mandato ALTER QMGR MQSC. Si utiliza la versión GSK_PCS11 del parámetro SSLCRYP, la etiqueta de la señal PKCS #11 debe especificarse enteramente en minúsculas.

El restablecimiento de claves secretas TLS y FIPS reciben soporte en IBM MQ MQI clients. Para obtener más información, consulte los temas “Restablecimiento de claves secretas SSL y TLS” en la página 475 y“Federal Information Processing Standards (FIPS) para AIX, Linux, and Windows” en la página 36.

Consulte “Configuración de la seguridad de IBM MQ MQI client” en la página 273 para obtener más información sobre el soporte TLS para los IBM MQ MQI clients.

Tareas relacionadas

Archivo de configuración de IBM MQ MQI client , `mqclient.ini`

Especificar que un canal MQI utiliza SSL/TLS

Para que un canal MQI utilice TLS, el valor del atributo *SSLCipherSpec*, del canal de conexión de cliente debe ser el nombre de un Ciphercliente IBM MQ en la plataforma de cliente.

Puede definir un canal de conexión de cliente con un valor para este atributo de las maneras siguientes. Se listan en el orden de prioridad descendente.

1. Cuando una salida Preconnect proporciona una estructura de definición de canal para utilizar.

   Una salida PreConnect puede proporcionar el nombre de un CipherSpec en el campo *SSLCipherSpec* de una estructura de definición de canal, MQCD. Esta estructura se devuelve en el campo **ppMQCDArrayPtr** de la estructura de parámetros de salida MQNXP utilizada por la salida PreConnect.
2. Cuando una aplicación cliente IBM MQ MQI client emite una llamada MQCONNX.

   La aplicación puede especificar el nombre de un CipherSpec en el campo *SSLCipherSpec* de una estructura de definición de canal, MQCD.Se hace referencia a esta estructura con la estructura de opciones de conexión MQCNO, que es un parámetro en la llamada MQCONNX.
3. Utilización de una tabla de definiciones de canal de cliente (CCDT).

   Una o varias entradas en una tabla de definiciones de canal de cliente pueden especificar el nombre de un CipherSpec.Por ejemplo, si crea una entrada mediante el mandato DEFINE CHANNEL MQSC, puede utilizar el parámetro SSLCIPH para especificar el nombre de un CipherSpec.
4. Utilización de Active Directory en Windows.

   En los sistemas Windows, puede utilizar el mandato de control **setmqscp** para publicar las definiciones de canal de conexión de cliente en Active Directory. Una o varias de estas definiciones pueden especificar el nombre de un CipherSpec.

Por ejemplo, si una aplicación cliente proporciona una definición de canal de conexión de cliente en una estructura MQCD en una llamada MQCONNX, esta definición se utilizará con preferencia a cualquier entrada de una tabla de definiciones de canal de cliente a la que el cliente IBM MQ puede acceder.

No puede utilizar la variable de entorno MQSERVER para proporcionar la definición de canal en el extremo cliente de un canal MQI que utiliza TLS.

Para comprobar si un certificado de cliente se ha transmitido, visualice el estado del canal en el extremo del servidor de un canal para saber si existe un valor de parámetro de nombre de igual.

Conceptos relacionados

"Especificación de una CipherSpec para un IBM MQ MQI client" en la página 452
Dispone de tres opciones para especificar una CipherSpec para un IBM MQ MQI client.

CipherSpecs y CipherSuites en IBM MQ

IBM MQ da soporte a TLS1.3 y TLS 1.2 CipherSpecs, y a algoritmos RSA y Diffie-Hellman. Sin embargo, puede habilitar las CipherSpecs en desuso, si tiene que hacerlo.

Consulte "Habilitación de CipherSpecs" en la página 428 si desea información sobre:

- CipherSpecs admitidas por IBM MQ.
- Cómo se habilitan las CipherSpecs SSL 3.0 y TLS 1.0. en desuso

IBM MQ da soporte a los algoritmos de intercambio y autenticación de claves RSA y Diffie-Hellman. El tamaño de la clave que se utiliza durante el reconocimiento TLS puede depender del certificado digital que se utiliza, pero algunas CipherSpecs incluyen una especificación del tamaño de clave de reconocimiento. Los tamaños de clave de reconocimiento más grandes proporcionan una autenticación más fuerte. Con los tamaños de clave más pequeños, el reconocimiento es más rápido.

Conceptos relacionados

"CipherSpecs y CipherSuites" en la página 22
Los protocolos de seguridad criptográficos deben estar de acuerdo con los algoritmos utilizados por una conexión segura. CipherSpecs y CipherSuites definen combinaciones específicas de algoritmos.

NSA Suite B Cryptography en IBM MQ

En este tema se proporciona información sobre cómo configurar IBM MQ for AIX, Linux, and Windows para que se ajuste al perfil TLS 1.2 compatible con Suite B.

Con el tiempo, el estándar NSA Cryptography Suite B Standard se ha ido actualizando para reflejar nuevos ataques contra los algoritmos y protocolos de cifrado. Por ejemplo, algunos CipherSpecs pueden dejar de certificarse con Suite B. Cuando se producen estos cambios, IBM MQ también se actualiza para implementar el último estándar. Como resultado, es posible que vea cambios en el comportamiento después de aplicar el mantenimiento. El archivo readme de IBM MQ muestra la versión de Suite B implementada por cada nivel de mantenimiento de producto. Si configura IBM MQ para implantar la conformidad con Suite B, consulte siempre el archivo readme cuando planifique el mantenimiento. Consulte IBM MQ, WebSphere MQ, y los archivos léame del producto MQSeries.

En sistemas AIX, Linux, and Windows , IBM MQ se puede configurar para que se ajuste al perfil TLS 1.2 compatible con Suite B en los niveles de seguridad que se muestran en la Tabla 1.

Tabla 3. Niveles de seguridad de Suite B con CipherSpecs y algoritmos de firma digital permitidos

Nivel de seguridad	**CipherSpecs permitidas**	**Algoritmos de firma digital permitidos**
128 bits	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA con SHA-256 ECDSA con SHA-384
192 bits	ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA con SHA-384

Tabla 3. Niveles de seguridad de Suite B con CipherSpecs y algoritmos de firma digital permitidos (continuación)		
Nivel de seguridad	**CipherSpecs permitidas**	**Algoritmos de firma digital permitidos**
Ambos [1]	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	ECDSA con SHA-256 ECDSA con SHA-384

1. Es posible configurar los niveles seguridad de 128 bits y 192 bits simultáneamente. Dado que la configuración de Suite B determina los algoritmos de cifrado mínimos aceptables, la configuración de ambos niveles de seguridad es equivalente a configurar sólo el nivel de seguridad de 128 bits. Los algoritmos de cifrado del nivel de seguridad de 192 bits son más fuertes que el mínimo necesario para el nivel de seguridad de 128 bits, por lo que están permitidos para el nivel de seguridad de 128 bits incluso si el nivel de seguridad de 192 bits no está habilitado.

Nota: Los convenios de denominación que se utilizan para el Nivel de seguridad no representan necesariamente el tamaño de la curva o elíptica del tamaño de la clave del algoritmo de cifrado AES.

Compatibilidad de CipherSpec con Suite B

Aunque el comportamiento predeterminado de IBM MQ no es cumplir con el estándar Suite B, IBM MQ se puede configurar para que se ajuste a uno o ambos niveles de seguridad en sistemas AIX, Linux, and Windows . Tras la configuración satisfactoria de IBM MQ para utilizar la Suite B, cualquier intento de iniciar un canal de salida utilizando un CipherSpec que no cumpla el estándar Suite B generará el error AMQ9282. Esta actividad también hace que el cliente MQI devuelva el código de razón MQRC_CIPHER_SPEC_NOT_SUITE_B. De forma similar, si se intenta iniciar un canal de entrada utilizando una CipherSpec que no se ajusta a la configuración de Suite B, se produce el error AMQ9616.

Para obtener más información sobre CipherSpecs de IBM MQ, consulte “Habilitación de CipherSpecs” en la página 428.

Suite B y los certificados digitales

Suite B limita los algoritmos de firma digital que se pueden utilizar para firmar certificados digitales. Suite B también restringe el tipo de clave pública que puede contener certificados. Por lo tanto, se debe haber configurado IBM MQ para que utilice los certificados cuyo algoritmo de firma digital y tipo de clave pública que permita el nivel de seguridad de Suite B configurado del socio remoto. Se rechazan los certificados digitales que no cumplan los requisitos del nivel de seguridad y la conexión falla con el error AMQ9633 o AMQ9285.

Para el nivel de seguridad de Suite B de 128 bits, se necesita la clave pública del asunto de certificado para poder utilizar la curva elíptica NIST P-256 o NIST P-384 y que se haya firmado con la curva elíptica NIST P-256 o la curva o elíptica NIST P-384. En el nivel de seguridad de Suite B de 192 bits, se necesita la clave pública del asunto de certificado para poder utilizar la curva elíptica NIST P-384, y que se haya firmado con la curva elíptica NIST P-384.

Para obtener un certificado adecuado que funcione de forma compatible con Suite B, utilice el mandato **runmqakm** y especifique el parámetro **-sig_alg** para solicitar un algoritmo de firma digital adecuado. Los valores de parámetro EC_ecdsa_with_SHA256 y EC_ecdsa_with_SHA384 **-sig_alg** corresponden a las claves de curva elíptica firmadas por los algoritmos de firma digital de Suite B permitidos.

Para obtener más información sobre el mandato **runmqakm**, consulte “Gestión de claves y certificados en AIX, Linux, and Windows” en la página 550.

Creación y solicitud de certificados digitales

Para crear un certificado digital autofirmado para probar Suite B, consulte “Creación de un certificado personal autofirmado en AIX, Linux, and Windows” en la página 551.

Para solicitar un certificado digital firmado por una CA para su utilización en la producción de Suite B, consulte “Solicitud de un certificado personal en AIX, Linux, and Windows” en la página 554.

Nota: La entidad emisora de certificados que se utilice deberá generar certificados digitales que cumplan los requisitos descritos en IETF RFC 6460.

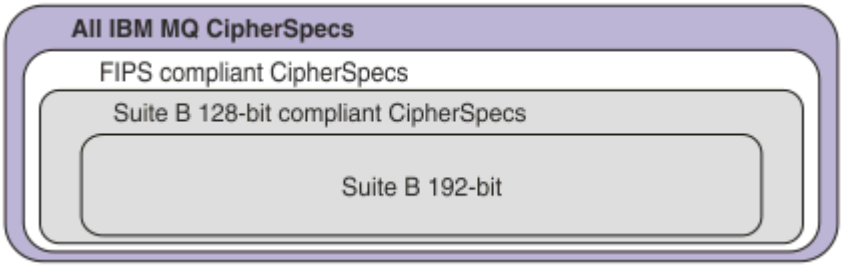
FIPS 140-2 y Suite B

Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC) . El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el certificado deIBM Crypto for C (ICC) y tener en cuenta cualquier consejo proporcionado por NIST. Un módulo FIPS 140-3 de sustitución está actualmente en curso y su estado se puede ver buscándolo en los módulos CMVP de NIST en la lista de procesos.

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los módulos CMVP de NIST en la lista de procesos.

El estándar Suite B es conceptualmente parecido a FIPS 140-2, ya que restringe el conjunto de algoritmos de cifrado permitidos para proporcionar un nivel de seguridad garantizado. Las CipherSpecs de Suite B soportadas actualmente se pueden utilizar cuando IBM MQ se ha configurado para que tenga un funcionamiento compatible con 140-2. Por consiguiente, es posible configurar IBM MQ para FIPS y Suite B de forma simultánea, en cuyo caso se aplican ambos conjuntos de restricciones.

El diagrama siguiente ilustra la relación entre estos subconjuntos:



Configuración de IBM MQ para el funcionamiento compatible con Suite B

Para obtener información sobre cómo configurar IBM MQ en AIX, Linux, and Windows para el funcionamiento compatible con Suite B, consulte “Configuración de IBM MQ para Suite B” en la página 45.

IBM MQ no da soporte al funcionamiento compatible con Suite B en las plataformas y clientes siguientes:

- Plataforma IBM i
- Plataforma z/OS
- Java cliente
- JMS cliente

Conceptos relacionados

“Especificación de que sólo se utilizan CipherSpecs certificadas por FIPS en el tiempo de ejecución del cliente MQI” en la página 274
Cree sus propios repositorios de claves utilizando software compatible con FIPS y, a continuación, especifique que el canal debe utilizar las CipherSpecs certificadas por FIPS.

ALW *Configuración de IBM MQ para Suite B*

IBM MQ se puede configurar para que funcione de conformidad con el estándar NSA Suite B en plataformas AIX, Linux, and Windows .

Suite B restringe el conjunto de algoritmos de cifrado permitidos para proporcionar un nivel de seguridad garantizado. IBM MQ se puede configurar para que funcione en conformidad con Suite B para proporcionar un nivel de seguridad mejorado. Para obtener más información sobre Suite B, consulte "Cifrado Suite B de la NSA (National Security Agency)" en la página 24. Para obtener más información sobre la configuración de Suite B y sus efectos sobre los canales TLS, consulte "NSA Suite B Cryptography en IBM MQ" en la página 43.

Gestor de colas

Para un gestor de colas, utilice el mandato **ALTER QMGR** con el parámetro **SUITEB** para establecer los valores adecuados para el nivel de seguridad que necesite. Para obtener más información, consulte ALTER QMGR.

También puede utilizar el mandato PCF **MQCMD_CHANGE_Q_MGR** con el parámetro **MQIA_SUITE_B_STRENGTH** para configurar el gestor de colas para que funcione de forma compatible con Suite B.

Nota: Si modifica valores de Suite B del gestor de colas, debe reiniciar el servicio MQXR para que estos valores entren en vigor.

Cliente MQI

De forma predeterminada, los clientes MQI no imponen la conformidad con Suite B. Puede habilitar la conformidad del cliente MQI para Suite B ejecutando una de las opciones siguientes:

1. Estableciendo el campo EncryptionPolicySuiteB en la estructura MQSCO en una llamada MQCONNX a uno o más de los valores siguientes:
   - MQ_SUITE_B_NONE
   - MQ_SUITE_B_128_BIT
   - MQ_SUITE_B_192_BIT

   El uso de MQ_SUITE_B_NONE con cualquier otro valor no es válido.

   Para obtener más información sobre la estructura MQSCO, consulte MQSCO-Opciones de configuración SSL.
2. Estableciendo la variable de entorno **MQSUITEB** en uno o varios de los valores siguientes:
   - NONE
   - 128_BIT
   - 192_BIT

   Puede especificar varios valores utilizando una lista separada por comas. El uso del valor NONE con cualquier otro valor no es válido.
3. Estableciendo el atributo **EncryptionPolicySuiteB** en la stanza SSL del archivo de configuración de cliente en uno o varios de los valores siguientes:
   - NONE
   - 128_BIT
   - 192_BIT

   Puede especificar varios valores utilizando una lista separada por comas. El uso de NONE con cualquier otro valor no es válido.

Nota: Los valores del cliente MQI se muestran en orden de prioridad. La estructura MSCO de la llamada MQCONNX altera temporalmente el valor de la variable de entorno **MQSUITEB** , que altera temporalmente el atributo de la stanza SSL.

.NET

Para los clientes no gestionados .NET, la propiedad **MQC.ENCRYPTION_POLICY_SUITE_B** indica el tipo de seguridad de Suite B necesaria.

Para obtener información sobre la utilización de Suite B en IBM MQ classes for .NET, consulte Clase MQEnvironment .NET.

AMQP

Los valores del atributo Suite B de un gestor de colas se aplican a los canales AMQP en dicho gestor de colas. Si modifica los valores de Suite B del gestor de colas, debe reiniciar el servicio AMQP para que los cambios entren en vigor.

Políticas de validación de certificados en IBM MQ

La política de validación de certificados determina controla el nivel de rigor con el que la validación de la cadena de certificados se ajusta a los estándares de la industria.

La política de validación de certificados depende de la plataforma y del entorno, tal como se indica a continuación:

- Para aplicaciones Java y JMS en todas las plataformas, la política de validación de certificados depende del componente JSSE del entorno de ejecución Java. Para obtener más información sobre la política de validación de certificados, consulte la documentación de su JRE.
- ALW Para los sistemas AIX, Linux, and Windows , la política de validación de certificados la proporciona IBM Global Security Kit (GSKit) y se puede configurar. V 9.4.0 V 9.4.0 Se admiten tres políticas de validación de certificados diferentes:
  - Una política de validación de certificados existente, utilizada para la máxima compatibilidad e interoperatividad con certificados digitales anteriores que no cumplan con los estándares de validación de certificados IETF actuales. Esta política se conoce como política Básica.
  - Una política de validación de certificados estricta y compatible con los estándares que impone el estándar RFC 5280. Esta política se conoce como política Estándar.
  - V 9.4.0 V 9.4.0 Una política de validación de certificados que no autentica el certificado de servidor TLS, disponible sólo para aplicaciones cliente.
- IBM i Para los sistemas IBM i, la política de validación de certificados depende de la biblioteca de sockets seguros proporcionada por el sistema operativo. Para obtener más información sobre la política de validación de certificados, consulte la documentación del sistema operativo.
- z/OS Para los sistemas z/OS, la política de validación de certificados depende del componente System SSL proporcionada por el sistema operativo. Para obtener más información sobre la política de validación de certificados, consulte la documentación del sistema operativo.

Para obtener información sobre cómo configurar la política de validación de certificados, consulte “Configuración de políticas de validación de certificados en IBM MQ” en la página 47. Para obtener más información sobre las diferencias entre las políticas de validación de certificados básica y estándar, consulte la sección Validación de certificados y diseño de políticas de confianza en AIX, Linux, and Windows.

Configuración de políticas de validación de certificados en IBM MQ

Existen varias formas en las que puede especificar qué política de validación de certificados TLS se utiliza para validar los certificados digitales recibidos de los sistemas asociados remotos.

Acerca de esta tarea

La política de validación de certificados determina controla el nivel de rigor con el que la validación de la cadena de certificados se ajusta a los estándares de la industria. La política de validación de certificados

depende de la plataforma y el entorno. Para obtener más información sobre las políticas de validación de certificados, consulte “Políticas de validación de certificados en IBM MQ” en la página 47.

Procedimiento

- Para establecer la política de validación de certificados en el gestor de colas, utilice el atributo de gestor de colas **CERTVPOL**.

  Para obtener más información sobre cómo establecer este atributo, consulte ALTER QMGR (modificar valores del gestor de colas).
- Para establecer la política de validación de certificados en el cliente, utilice los métodos siguientes.

  Si se utiliza más de un método para establecer la política, el cliente utiliza los valores en el siguiente orden de prioridad:

  1. Utilice el campo `CertificateValPolicy` en la estructura MQSCO del cliente. Establezca el campo en uno de los valores siguientes:

     MQ_CERT_VAL_POLICY_ANY
     Aplique cada una de las políticas de validación de certificados soportadas por la biblioteca de sockets seguros. Acepte la cadena de certificados si alguna de las políticas considera válida la cadena de certificados.

     MQ_CERT_VAL_POLICY_RFC5280
     Aplique sólo la política de validación de certificados compatible con RFC5280 . Este valor proporciona una validación más estricta que el valor ANY, pero rechaza algunos certificados digitales más antiguos.

     V 9.4.0 V 9.4.0 **MQ_CERT_VAL_POLICY_NONE**
     No aplicar ninguna política de validación de certificados. Este valor es solo para aplicaciones cliente y acepta el certificado de servidor TLS sin validar la cadena de confianza.

     Si desea más información sobre cómo utilizar este campo, consulte MQSCO - opciones de configuración SSL.
  2. Utilice la variable de entorno de cliente **MQCERTVPOL**. Para establecer esta variable de entorno, utilice uno de los mandatos siguientes:

     – 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. Utilice el atributo **CertificateValPolicy** de la stanza SSL en el archivo de configuración del cliente. Establezca este atributo en uno de los valores siguientes:

     CUALQUIERA
     Utilizar cualquier política de validación de certificados soportada por la biblioteca de sockets seguros subyacente. Este valor es el predeterminado.

     RFC5280
     Utilizar sólo la validación de certificados que cumpla con el estándar RFC 5280.

V 9.4.0 V 9.4.0 **NINGUNO**

No aplicar ninguna política de validación de certificados. Este valor acepta el certificado de servidor TLS sin validar la cadena de confianza.

Para obtener más información sobre cómo utilizar este atributo, consulte Stanza SSL del archivo de configuración de cliente.

Certificados digitales y compatibilidad de CipherSpec en IBM MQ

En este tema se proporciona información sobre cómo elegir las CipherSpecs y los certificados digitales adecuados para su política de seguridad, describiendo la relación entre las CipherSpecs y los certificados digitales en IBM MQ.

Únicamente un subconjunto de las CipherSpecs soportadas puede utilizarse con todos los tipos de certificados digitales soportados. Por consiguiente, es necesario que elija una CipherSpec adecuada para su certificado digital. Del mismo modo, si la política de seguridad de la organización requiere que utilice una CipherSpec determinada, debe obtener un certificado digital apropiado para dicha CipherSpec.

El algoritmo de firmas digitales MD5 y TLS 1.2

Los certificados digitales firmados mediante el algoritmo MD5 se rechazan cuando se utiliza el protocolo TLS 1.2. Esto se debe a que, ahora, muchos analistas consideran que el algoritmo MD5 es débil y, en general, se desaconseja su uso. Para utilizar CipherSpecs basadas en el protocolo TLS 1.2, asegúrese de que los certificados digitales no utilicen el algoritmo MD5 y sus firmas digitales. Las CipherSpecs que utilizan los protocolos TLS 1.0 no están sujetos a esta restricción y pueden continuar utilizando certificados con firmas digitales MD5.

Para ver el algoritmo de firma digital para un certificado determinado, puede utilizar el mandato **runmqakm** :

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

donde *etiqueta_certificado* es la etiqueta del certificado cuyo algoritmo de firma digital se ha de visualizar. Para obtener información detallada, consulte Etiquetas de certificados digitales.

La ejecución del mandato **runmqakm** genera una salida que muestra el uso del algoritmo de firma 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
```

La línea `Signature Algorithm` muestra que se utiliza el algoritmo `MD5WithRSASignature` . Este algoritmo se basa en MD5 y por lo tanto este certificado digital no se puede utilizar con las CipherSpecs de TLS 1.2.

Interoperatividad de Elliptic Curve y CipherSpecs RSA

No se puede utilizar todas las CipherSpecs con todos los certificados digitales. Las CipherSpecs se indican mediante el prefijo de nombre CipherSpec . Cada tipo de CipherSpec impone diferentes restricciones sobre el tipo de certificado digital que se puede utilizar. Estas restricciones se aplican a todas las conexiones TLS de IBM MQ, pero resultan especialmente relevantes para los usuarios del cifrado Elliptic Curve.

La siguiente tabla resume las relaciones entre las CipherSpecs y los certificados digitales:

Tabla 4. Relaciones entre las CipherSpecs y los certificados digitales

Tipo	Prefijo de nombre de CipherSpec	Descripción	Tipo de clave pública necesaria	Algoritmo de cifrado de firma digital	Método de establecimiento de claves secretas
1	ECDHE_ECDSA_	CipherSpecs que utilizan claves públicas Elliptic Curve, claves secretas Elliptic Curve y algoritmos de firma digital Elliptic Curve.	Elliptic Curve	ECDSA	ECDHE
2	ECDHE_RSA_	CipherSpecs que utilizan claves públicas RSA, claves secretas de 4Elliptic Curve y algoritmos de firma digital RSA.	RSA	RSA	ECDHE
3	(All TLS 1.3 CipherSpecs)	CipherSpecs que utilizan claves públicas Elliptic Curve o RSA, claves secretas Elliptic Curve y algoritmos de firma digital Elliptic Curve o RSA.	Curva elíptica o RSA	ECDSA o RSA	ECDHE o RSA
4	(Todos los demás)	CipherSpecs que utilizan claves públicas RSA y algoritmos de firma digital RSA.	RSA	RSA	RSA

Nota: Los gestores de colas IBM MQ y los clientes MQI sólo dan soporte a las CipherSpecs de Tipo 1 y 2 en las plataformas IBM i.

En la columna de tipo de clave pública necesaria se muestra el tipo de clave pública que el certificado personal debe tener cuando utiliza cada tipo de CipherSpec. El certificado personal es el certificado de entidad final que identifica al gestor de colas o al cliente ante su socio remoto.

Debe asegurarse de que el certificado que se nombra en la etiqueta de certificado es adecuado para el canal CipherSpec. Es decir, si configura un canal con una CipherSpec que requiere un certificado Elliptic

Curve (EC), no puede nombrar un certificado RSA en la etiqueta del certificado. Si configura un canal con una CipherSpec que requiere un certificado RSA, no puede nombrar un certificado EC en la etiqueta del certificado.

Presuponiendo que ha configurado correctamente IBM MQ, puede tener:

- Un gestor de colas individual con una combinación de certificados RSA y EC.
- Diferentes canales en el mismo gestor de colas que utilizan un certificado RSA o un certificado EC.

El algoritmo de cifrado de firma digital hace referencia al algoritmo de cifrado que se utiliza para validar al igual. El algoritmo de cifrado se utiliza junto con un algoritmo de hash como, por ejemplo, MD5, SHA-1 o SHA-256 para calcular la firma digital. Existen distintos algoritmos de firma digital que se pueden utilizar, por ejemplo, RSA con MD5 o ECDSA con SHA-256. En la tabla, ECDSA hace referencia al conjunto de algoritmos de firma digital que utilizan ECDSA; RSA hace referencia al conjunto de algoritmos de firma digital que utilizan RSA. Se puede utilizar cualquier algoritmo de firma digital soportado en el conjunto, siempre que se base en el algoritmo de cifrado indicado.

Las CipherSpecs de Tipo 1 requieren que el certificado personal tenga una clave pública Elliptic Curve. Cuando se utilizan estas CipherSpecs, se utiliza el acuerdo de claves Elliptic Curve Diffie Hellman Ephemeral para establecer la clave secreta de la conexión.

Las CipherSpecs de Tipo 2 requieren que el certificado personal tenga una clave pública RSA. Cuando se utilizan estas CipherSpecs, se utiliza el acuerdo de claves Elliptic Curve Diffie Hellman Ephemeral para establecer la clave secreta de la conexión.

Las CipherSpecs de tipo 3 requieren que el certificado personal tenga una clave pública RSA. Cuando se utilizan estas CipherSpecs, se utiliza el intercambio de claves RSA para establecer la clave secreta de la conexión.

Esta lista de restricciones no es exhaustiva: dependiendo de la configuración, puede haber restricciones adicionales que pueden afectar aún más a la capacidad de interoperar. Por ejemplo, si IBM MQ se ha configurado para cumplir los estándares FIPS 140-2 o Suite B de la NSA , esto también limitará el rango de configuraciones permitidas. Para obtener más información, consulte el siguiente apartado.

Si necesita utilizar diferentes tipos de CipherSpec en el mismo gestor de colas o aplicación de cliente, configure una combinación de etiqueta de certificado y CipherSpec en la definición de cliente.

Los tres tipos de CipherSpec no interactúan directamente: se trata de una limitación de los estándares actuales de TLS. Por ejemplo, supongamos que ha elegido utilizar ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec para un canal receptor denominado TO.QM1 en un gestor de colas denominado QM1, el receptor debe tener un certificado personal con una clave Elliptic Curve y una firma digital basada en ECDSA. Si el canal receptor no cumple estos requisitos, el canal no se inicia.

Otros canales que se conecten al gestor de colas QM1 pueden utilizar otras CipherSpecs, siempre que cada canal utilice un certificado del tipo correcto para la CipherSpec de dicho canal. Por ejemplo, presuponga que QM1 utiliza un canal emisor denominado TO.QM2 para enviar mensajes a otro gestor de colas denominado QM2. El canal TO.QM2 puede utilizar la CipherSpec de Tipo 3 TLS_RSA_WITH_AES_256_CBC_SHA256 siempre que ambos extremos del canal utilicen certificados que contengan claves públicas RSA. Se puede utilizar el atributo de canal de etiqueta para configurar un certificado diferente para cada canal.

Cuando planifique sus redes IBM MQ, considere detenidamente qué canales requieren TLS y asegúrese de que el tipo de los certificados utilizados para cada canal sea adecuado para su uso con la CipherSpec en dicho canal.

Para ver el algoritmo de firma digital y el tipo de clave pública de un certificado digital, puede utilizar el mandato **runmqakm**:

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

donde *etiqueta_certificado* es la etiqueta del certificado cuyo algoritmo de firma digital necesita visualizar. Para obtener información detallada, consulte Etiquetas de certificados digitales.

La ejecución del mandato **runmqakm** generará una salida que muestra el tipo de clave 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
```

En la línea Tipo de clave pública de este caso se muestra que el certificado tiene una clave pública Elliptic Curve. En la línea Algoritmo de firma de este caso se muestra que el algoritmo EC_ecdsa_with_SHA384 está en uso: se basa en el algoritmo ECDSA. Por tanto, este certificado sólo resulta adecuado para utilizarlo con las CipherSpecs de Tipo 1.

TLS 1.3 CipherSpecs

TLS 1.3 CipherSpecs da soporte a los certificados ECDSA y RSA.

CipherSpecs Elliptic Curve y Suite B de la NSA

Cuando se configura IBM MQ conforme al perfil TSL 1.2 compatible con Suite B, las CipherSpecs permitidas y los algoritmos de firma digital se restringen, tal como se describe en “NSA Suite B Cryptography en IBM MQ” en la página 43. Adicionalmente, el rango de claves Elliptic Curve aceptable se reduce, según los niveles de seguridad configurados.

En el nivel de seguridad de Suite B de 128 bits, se necesita la clave pública del asunto de certificado para poder utilizar la curva elíptica NIST P-256 o NIST P-384 y que se haya firmado con la curva elíptica NIST P-256 o la curva o elíptica NIST P-384. Se puede utilizar el mandato **runmqakm** para solicitar certificados digitales para este nivel de seguridad utilizando un parámetro -sig_alg de EC_ecdsa_with_SHA256 o EC_ecdsa_with_SHA384.

En el nivel de seguridad de Suite B de 192 bits, se necesita la clave pública del asunto de certificado para poder utilizar la curva elíptica NIST P-384, y que se haya firmado con la curva elíptica NIST P-384. Se puede utilizar el mandato **runmqakm** para solicitar certificados digitales para este nivel de seguridad utilizando un parámetro -sig_alg de EC_ecdsa_with_SHA384.

Las curvas elípticas NIST a las que se da soporte son las siguientes:

Tabla 5. Curvas elípticas NIST a las que se da soporte		
Nombre de curva NIST FIPS 186-3	**Nombre de curva RFC 4492**	**Tamaño de clave de Elliptic Curve (bits)**
P-256	secp256r1	256
P-384	secp384r1	384
P-521	secp521r1	521

Nota: La curva elíptica NIST P-521 no se puede utilizar para el funcionamiento compatible con Suite B.

Conceptos relacionados

"Habilitación de CipherSpecs" en la página 428
Habilite una CipherSpec utilizando el parámetro **SSLCIPH** en el mandato **DEFINE CHANNEL** o **ALTER CHANNEL** MQSC.

"Especificación de que sólo se utilizan CipherSpecs certificadas por FIPS en el tiempo de ejecución del cliente MQI" en la página 274
Cree sus propios repositorios de claves utilizando software compatible con FIPS y, a continuación, especifique que el canal debe utilizar las CipherSpecs certificadas por FIPS.

"NSA Suite B Cryptography en IBM MQ" en la página 43
En este tema se proporciona información sobre cómo configurar IBM MQ for AIX, Linux, and Windows para que se ajuste al perfil TLS 1.2 compatible con Suite B.

"Cifrado Suite B de la NSA (National Security Agency)" en la página 24
El gobierno de los Estados Unidos de América ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. La NSA (National Security Agency) de Estados Unidos recomienda un conjunto de algoritmos de cifrado interoperables en su estándar Suite B.

Registros de autenticación de canal

Para ejercer un control más preciso sobre el acceso otorgado a la conexión de sistemas a nivel de canal, puede utilizar registros de autenticación de canal.

Un cliente puede intentar conectarse al gestor de colas utilizando un ID de usuario en blanco o un ID de usuario de alto nivel que permita al cliente realizar acciones malintencionadas. Puede bloquear el acceso a estos clientes utilizando registros de autenticación de canal. O bien, un cliente puede declarar un ID de usuario que sea válido en la plataforma del cliente, pero que sea desconocido o tenga un formato no válido en la plataforma del servidor. Puede utilizar un registro de autenticación de canal para correlacionar el ID de usuario declarado para un ID de usuario válido.

Puede encontrar una aplicación cliente que se conecta al gestor de colas y se comporta mal de alguna manera. Para proteger el servidor frente a los problemas que esta aplicación está causando, es necesario bloquearla temporalmente utilizando la dirección IP de la aplicación cliente hasta que se actualicen las reglas del cortafuegos o se corrija la aplicación cliente. Puede utilizar un registro de autenticación de canal para bloquear la dirección IP desde la que se conecta la aplicación cliente.

Si ha configurado una herramienta de administración tal como IBM MQ Explorer, y un canal para ese uso específico, puede asegurarse de que sólo puedan utilizarlo sistemas clientes determinados. Puede utilizar un registro de autenticación de canal para que el canal sólo pueda ser utilizado desde direcciones IP determinadas.

Si acaba de empezar con algunas aplicaciones de ejemplo que se ejecutan como cliente, consulte Preparación y ejecución de los programas de ejemplo para obtener un ejemplo de configuración del gestor de colas de forma segura utilizando registros de autenticación de canal.

Si desea obtener registros de autenticación de canal para controlar canales de entrada, utilice el mandato de MQSC **ALTER QMGR CHLAUTH(ENABLED)**.

Se aplican las reglas **CHLAUTH** para un MCA de canal que se crea en respuesta a una nueva conexión de entrada. Para un MCA de canal creado en respuesta al canal que se está iniciando localmente, no se aplica ninguna regla **CHLAUTH**.

Tabla 6. Dónde se aplican las reglas CHLAUTH para diferentes pares de canales

Tipo de canal	**MCA donde se aplican las reglas CHLAUTH**
SDR-RCVR	RCVR
RQSTR-SVR (iniciado en SVR)	RQSTR
RQSTR-SVR (iniciado en RQSTR)	SVR
RQSTR-SDR (iniciado en SDR)	RQSTR
RQSTR-SDR (iniciado en RQSTR)	SDR para la conexión inicial. RQSTR para la conexión de devolución de llamada.

Se pueden crear registros de autenticación de canal para realizar las funciones siguientes:

- Bloquear conexiones realizadas desde direcciones IP específicas.
- Bloquear conexiones realizadas desde identificadores de usuario específicos.
- Definir un valor MCAUSER para ser utilizado para cualquier canal que se conecte desde una dirección IP determinada.
- Definir un valor MCAUSER para ser utilizado para cualquier canal que declare un ID de usuario determinado.
- Definir un valor MCAUSER para ser utilizado para cualquier canal que tenga un determinado nombre distinguido de SSL o TLS.
- Definir un valor MCAUSER para ser utilizado para cualquier canal que se conecte desde un gestor de colas determinado.
- Bloquear conexiones que declaren proceder de un gestor de colas determinado, a menos que la conexión proceda de una dirección IP específica.
- Bloquear conexiones que presenten un certificado SSL o TLS determinado, a menos que la conexión proceda de una dirección IP específica.

Estos usos se explican con más detalle en las secciones siguientes.

Puede crear, modificar o eliminar registros de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH** o el mandato PCF **Set Channel Authentication Record**.

Nota: Un gran número de registros de autenticación de canal puede tener un impacto negativo en el rendimiento de un gestor de colas.

Bloqueo de direcciones IP

La función normal de un cortafuegos es impedir el acceso desde determinadas direcciones IP. No obstante, puede haber ocasiones en que se produzcan intentos de conexión desde una dirección IP que no debería tener acceso a su sistema IBM MQ y deberá bloquear la dirección temporalmente para que se pueda actualizar el cortafuegos. Es posible que estos intentos de conexión no procedan de canales de IBM MQ; estos intentos de conexión puede que procedan de otras aplicaciones de socket que están mal configuradas para dirigirse a su escucha de IBM MQ. Puede bloquear direcciones IP estableciendo un registro de autenticación de canal de tipo BLOCKADDR. Puede especificar una o más direcciones individuales, rangos de direcciones, o patrones que incluyan caracteres comodín.

Cada vez que se rechaza una conexión de entrada porque la dirección IP se ha bloqueado de esta manera, se emite un mensaje de suceso MQRC_CHANNEL_BLOCKED con el calificador de razón MQRQ_CHANNEL_BLOCKED_ADDRESS, siempre que los sucesos de canal estén habilitados y el gestor de colas esté en ejecución. Además, la conexión se mantiene abierta durante 30 segundos antes de devolver el error para asegurar que el escucha no se vea inundado con repetidos intentos de conexión que están bloqueados.

Para bloquear direcciones IP sólo en canales específicos, o para evitar el retardo antes de notificar el error, establezca un registro de autenticación de canal de tipo ADDRESSMAP con el parámetro USERSRC(NOACCESS).

Cada vez que se rechaza una conexión de entrada por esta razón, se emite un mensaje de suceso MQRC_CHANNEL_BLOCKED con el calificador de razón MQRQ_CHANNEL_BLOCKED_NOACCESS, siempre que los sucesos de canal estén habilitados y el gestor de colas esté en ejecución.

Consulte "Bloquear direcciones IP específicas" en la página 391 para obtener un ejemplo.

Bloqueo de los ID de usuario

Para evitar que determinados identificadores de usuario se conecten a través de un canal de cliente, establezca un registro de autenticación de canal de tipo BLOCKUSER. Este tipo de registro de autenticación de canal se aplica sólo a los canales de cliente, no a los canales de mensajes. Puede especificar uno o más identificadores de usuario para bloquear, pero no puede utilizar comodines.

Cada vez que se rechaza una conexión de entrada por esta razón, se emite un mensaje de suceso MQRC_CHANNEL_BLOCKED con el calificador de razón MQRQ_CHANNEL_BLOCKED_USERID, siempre que los sucesos de canal estén habilitados.

Consulte "Bloquear identificadores (ID) de usuario específicos" en la página 393 para obtener un ejemplo.

Puede también bloquear cualquier acceso para identificadores de usuario especificados y determinados canales estableciendo un registro de autenticación de canal de tipo USERMAP mediante el parámetro USERSRC(NOACCESS).

Cada vez que se rechaza una conexión de entrada por esta razón, se emite un mensaje de suceso MQRC_CHANNEL_BLOCKED con el calificador de razón MQRQ_CHANNEL_BLOCKED_NOACCESS, siempre que los sucesos de canal estén habilitados y el gestor de colas esté en ejecución.

Consulte "Bloqueo del acceso de un ID de usuario cliente" en la página 396 para obtener un ejemplo.

Bloqueo de nombres de gestores de colas

Para bloquear el acceso a cualquier canal que se conecte desde un gestor de colas especificado, establezca un registro de autenticación de canal de tipo QMGRMAP con el parámetro USERSRC(NOACCESS). Puede especificar un nombre de gestor de colas individual o un patrón de caracteres que incluya comodines. No existe ningún homólogo de la función BLOCKUSER para bloquear el acceso para gestores de colas.

Cada vez que se rechaza una conexión de entrada por esta razón, se emite un mensaje de suceso MQRC_CHANNEL_BLOCKED con el calificador de razón MQRQ_CHANNEL_BLOCKED_NOACCESS, siempre que los sucesos de canal estén habilitados y el gestor de colas esté en ejecución.

Consulte "Bloquear el acceso desde un gestor de colas remoto" en la página 396 para obtener un ejemplo.

Bloqueo de nombres distinguidos de SSL o TLS

Para bloquear el acceso a cualquier usuario que declare un certificado personal SSL o TLS que contenga un nombre distinguido especificado, establezca un registro de autenticación de canal de tipo SSLPEERMAP con el USERSRC(NOACCESS). Puede especificar un nombre distinguido individual o un patrón de caracteres que incluya comodines. No existe ningún homólogo de la función BLOCKUSER para bloquear el acceso para nombres distinguidos.

Cada vez que se rechaza una conexión de entrada por esta razón, se emite un mensaje de suceso MQRC_CHANNEL_BLOCKED con el calificador de razón MQRQ_CHANNEL_BLOCKED_NOACCESS, siempre que los sucesos de canal estén habilitados y el gestor de colas esté en ejecución.

Consulte "Bloquear el acceso para un Nombre distinguido SSL o TLS" en la página 397 para obtener un ejemplo.

Correlación de direcciones IP con los ID de usuario que se deben utilizar

Para especificar que cualquier canal que se conecte desde una dirección IP especificada debe utilizar un MCAUSER específico, establezca un registro de autenticación de canal de tipo ADDRESSMAP. Puede especificar una dirección individual, un rango de direcciones, o un patrón de caracteres que incluya comodines.

Si utiliza un reenviador de puertos, interruptor de sesión DMZ, o cualquier otra configuración que cambie la dirección IP presentada al gestor de colas, la correlación de direcciones IP puede no ser adecuada en su caso.

Consulte “Correlacionar una dirección IP con un ID de usuario MCAUSER” en la página 397 para obtener un ejemplo.

Correlación nombres de gestores colas con los ID de usuario que se deben utilizar

Para especificar que cualquier canal que se conecte desde un gestor de colas especificado debe utilizar un MCAUSER específico, establezca un registro de autenticación de canal de tipo QMGRMAP. Puede especificar un nombre de gestor de colas individual o un patrón de caracteres que incluya comodines.

Consulte “Correlacionar un gestor de colas remoto con un ID de usuario MCAUSER” en la página 394 para obtener un ejemplo.

Correlación de los ID de usuario declarados por un cliente con los ID de usuario que se deben utilizar

Para especificar que si una conexión de cliente de IBM MQ MQI utiliza un determinado ID de usuario debe utilizar un MCAUSER diferente especificado, establezca un registro de autenticación de canal de tipo USERMAP. La correlación de identificadores de usuario no utiliza comodines.

Consulte “Correlación de un ID de usuario cliente con un ID de usuario MCAUSER” en la página 394 para obtener un ejemplo.

Correlación de nombres distinguidos de SSL o TLS con los ID de usuario que se deben utilizar

Para especificar que cualquier usuario que declare un certificado personal SSL/TLS que contenga un nombre distinguido especificado debe utilizar un MCAUSER específico, establezca un registro de autenticación de canal de tipo SSLPEERMAP. Puede especificar un nombre distinguido individual o un patrón de caracteres que incluya comodines.

Consulte “Correlacionar un Nombre distinguido SSL o TLS con un ID de usuario MCAUSER” en la página 395 para obtener un ejemplo.

Correlación de gestores de colas, clientes o SSL o TLS DN de acuerdo con las direcciones IP

En algunos casos, un tercero puede suplantar un nombre de gestor de colas. También puede ser robado y reutilizado un certificado SSL o TLS o un archivo de base de datos de claves. Para protegerse contra estas amenazas, puede especificar que una conexión procedente de un gestor de colas o cliente determinado, o que utilice un nombre distinguido determinado se debe conectar desde una dirección IP especificada. Establezca un registro de la autenticación de canal de tipo USERMAP, QMGRMAP o SSLPEERMAP y especifique la dirección IP permitida, o patrón de direcciones IP permitidas, utilizando el parámetro ADDRESS.

Consulte “Correlacionar un gestor de colas remoto con un ID de usuario MCAUSER” en la página 394 para obtener un ejemplo.

Interacción entre registros de autenticación de canal

Es posible que un canal que intenta establecer una conexión coincida con más de un registro de autenticación de canal, y que estos registros tengan efectos contradictorios. Por ejemplo, un canal puede declarar un ID de usuario que está bloqueado por un registro de autenticación canal BLOCKUSER, pero

con un certificado SSL o TLS que coincida con un registro SSLPEERMAP que define un ID de usuario diferente. Además, si los registros de autenticación de canal utilizan comodines, una dirección IP, nombre de gestor de colas o nombre distinguido de SSL o TLS puede coincidir con varios patrones de caracteres. Por ejemplo, la dirección IP 192.0.2.6 coincide con los patrones 192.0.2.0-24, 192.0.2.*, y 192.0.*.6. La acción emprendida se determina de la forma siguiente.

- El registro de autenticación de canal utilizado se selecciona de la manera siguiente:
  - Un registro de autenticación de canal que coincida explícitamente con el nombre de canal tiene prioridad sobre un registro de autenticación de canal que coincida con el nombre de canal utilizando un comodín.
  - Un registro de autenticación de canal que haga uso de un nombre distinguido de SSL o TLS tiene prioridad sobre un registro que haga uso de un ID de usuario, nombre de gestor de colas o dirección IP.
  - Un registro de autenticación de canal que haga uso de un ID de usuario o nombre de gestor de colas tiene prioridad sobre un registro que haga uso de una dirección IP.
- Si se encuentra un registro de autenticación de canal coincidente y éste especifica un MCAUSER, este MCAUSER se asigna al canal.
- Si se encuentra un registro de autenticación de canal coincidente y éste especifica que el canal no tiene acceso, se asigna al canal un MCAUSER con el valor *NOACCESS. Este valor puede ser cambiado más tarde por un programa de salida de seguridad.
- Si no se encuentra ningún registro de autenticación de canal coincidente o se encuentra un registro coincidente y especifica que se debe utilizar el ID de usuario del canal, se examina el campo MCAUSER.
  - Si el campo MCAUSER está en blanco, se asigna el ID de usuario del cliente al canal.
  - Si el campo MCAUSER no está en blanco, su valor se asigna al canal.
- Se ejecuta cualquier programa de salida de seguridad. Este programa de salida puede establecer el ID de usuario del canal o determinar que se debe bloquear el acceso.
- Si se bloquea la conexión o MCAUSER está establecido en *NOACCESS, se cierra el canal.
- Si la conexión no se bloquea, para cualquier canal excepto un canal de cliente, se compara el ID de usuario de canal determinado en los pasos anteriores con la lista de usuarios bloqueados.
  - Si el ID de usuario está en la lista de usuarios bloqueados, el canal se cierra.
  - Si el ID de usuario no está en la lista de usuarios bloqueados, el canal se ejecuta.

Cuando varios registros de autenticación de canal coinciden con un nombre de canal, una dirección IP, un nombre de host, un nombre de gestor de colas o el nombre distinguido de SSL o TLS, se utiliza la coincidencia más específica. La coincidencia que se considera como:

- La más específica, es un nombre sin caracteres comodín, por ejemplo:
  - Un nombre de canal de A.B.C
  - Una dirección IP de 192.0.2.6
  - Un nombre de host de `hursley.ibm.com`
  - Un nombre de gestor de colas de 192.0.2.6
- La más genérica, es un solo asterisco (*) coincidente, por ejemplo:
  - Todos los nombres de canal
  - Todas las direcciones IP
  - Todos los nombres de host
  - Todos los nombres de gestores de colas
- Un patrón con un asterisco al principio de una serie es más genérico que un valor definido al principio de una serie:
  - Para los canales, *.B.C es más genérico que A.*
  - Para las direcciones IP, *.0.2.6 es más genérico que 192.*

  - Para los nombres de host, `*.ibm.com` es más genérico que `hursley.*`
  - Para los nombres de gestores de colas, *QUEUEMANAGER es más genérico que QUEUEMANAGER*
- Un patrón con un asterisco en un lugar específico en una serie es más genérico que un valor definido en el mismo lugar de una serie y, del mismo modo, para cada lugar posterior de una serie:
  - Para los canales, A.*.C es más genérico que A.B.*
  - Para las direcciones IP, 192.*.2.6 es más genérico que 192.0.*.
  - Para los nombres de host, `hursley.*.com` es más genérico que `hursley.ibm.*`
  - Para los nombres de gestores de colas, Q*MANAGER es más genérico que QUEUE*
- Cuando dos o más patrones tienen un asterisco en un lugar específico de una serie, el que tiene menos nodos después del asterisco es más genérico:
  - Para canales, A.* es más genérico que A.*.C
  - Para direcciones IP, 192.* es más genérico que 192.*.2.*.
  - Para los nombres de host, `hurlsey.*` es más genérico que `hursley.*.com`
  - Para los nombres de gestores de colas, Q* es más genérico que Q*MGR
- Adicionalmente, para una dirección IP:
  - Un rango indicado con un guión (-) es más específico que un asterisco. Por tanto, 192.0.2.0-24 es más específico que 192.0.2.*.
  - Un rango que es un subconjunto de otro mayor es más específico que el rango mayor. Por tanto, 192.0.2.5-15 es más específico que 192.0.2.0-24.
  - No están permitidos los rangos solapados. Por ejemplo, no puede tener registros de autenticación de canal para 192.0.2.0-15 y 192.0.2.10-20 al mismo tiempo.
  - Un patrón no puede tener menos números de componentes que los necesarios a no ser que el patrón termine con un asterisco individual final. Por ejemplo, 192.0.2 no es válido, pero 192.0.2.* es válido.
  - Un asterisco final debe separarse del resto de la dirección mediante el separador de parte adecuado (un punto (.) para IPv4, dos puntos (:) para IPv6). Por ejemplo, 192.0* no es válido porque el asterisco no está separado.
  - Un patrón puede contener asteriscos adicionales, a condición de que no haya ningún asterisco adyacente al asterisco final. Por ejemplo, 192.*.2.* es válido, pero 192.0.*.* no es válido.
  - Un patrón de dirección IPv6 no puede contener un signo doble de dos puntos y un asterisco final, porque la dirección resultante será ambigua. Por ejemplo, 2001::* podría expandirse a 2001:0000:*, 2001:0000:0000:* etc.
- Para un nombre distinguido de SSL o TLS, el orden de prioridad de las subseries de caracteres es el siguiente:

Tabla 7. Orden de prioridad de subseries

Orden	**Subserie de nombre distinguido**	**Nombre**
1	SERIALNUMBER=	Número de serie de certificado
2	MAIL=	Dirección de correo electrónico
3	Deprecated E=	Dirección de correo electrónico (En desuso por ser preferible MAIL)
4	UID=, USERID=	Identificador de usuario
5	CN=	Nombre común
6	T =	Título

Tabla 7. Orden de prioridad de subseries (continuación)		
Orden	**Subserie de nombre distinguido**	**Nombre**
7	OU=	Unidad organizativa
8	DC=	Componente de dominio
9	O=	Organización
10	STREET=	Calle / Primera línea de dirección
11	L=	Localidad
12	ST=, SP=, S=	Nombre del estado o provincia
13	P=	Código postal
14	C=	País
15	UNSTRUCTUREDNAME=	Nombre de host
16	UNSTRUCTUREDADDRESS=	Dirección IP
17	DNQ=	Calificador de nombre distinguido

Por tanto, si un certificado SSL o TLS se presenta con un DN que contenga las subseries O=IBM y C=UK, IBM MQ utiliza preferentemente un registro de autenticación de canal para O=IBM, en vez de uno para C=UK, si ambos están presentes.

Un nombre distinguido puede contener varias OU, que se deben especificar en orden jerárquico con las unidades organizativas más grandes especificadas primero. Si dos nombres distinguidos son iguales en todos los sentidos excepto por sus valores de unidad organizativa, el nombre distinguido más específico se determina de la siguiente manera:

1. Si tienen diferentes números de atributos de unidad organizativa, el nombre distinguido con más valores de unidad organizativa es más específico. La razón es que el nombre distinguido con más unidades organizativas califica más en detalle al nombre distinguido y proporciona más criterios de coincidencia. Aunque la unidad organizativa de nivel superior fuera un asterisco (OU=*), el nombre distinguido con más unidades organizativas sigue considerándose como el más específico.
2. Si tienen el mismo número de atributos de unidad organizativa, los pares correspondientes de valores de unidad organizativa se comparan en secuencia, de izquierda a derecha, donde la unidad organizativa más a la izquierda es el nivel superior (menos específica), de acuerdo con las reglas siguientes:
   a. Una unidad organizativa sin valores de asterisco es la más específica porque sólo puede coincidir con una serie.
   b. Una unidad organizativa con un único asterisco al principio o al final (por ejemplo, OU=ABC* o OU=*ABC) es la siguiente más específica.
   c. Una unidad organizativa con dos asteriscos, (por ejemplo OU=*ABC*) es la siguiente más específica.
   d. Una unidad organizativa formada sólo por un asterisco (OU=*) es la menos específica.
3. Si la comparación de series es entre dos valores de atributo de la misma especificidad, la serie del atributo más largo es más específica.
4. Si la comparación de series es entre dos valores de atributo de la misma especificidad y longitud, se comparan las series (sin tener en cuenta mayúsculas y minúsculas) de la parte del nombre distinguido excluidos los asteriscos.

Si dos nombres distinguidos son iguales en todos los aspectos excepto en sus valores de DC, se aplican las mismas reglas de coincidencia que para las OU, excepto que en los valores de DC, el DC más izquierda es el nivel más bajo (más específico) y el orden de comparación difiere en consecuencia.

Visualización de registros de autenticación de canal

Para visualizar registros de autenticación de canal, utilice el mandato MQSC **DISPLAY CHLAUTH** o el mandato PCF **Inquire Channel Authentication Records**. Puede obtener todos los registros que coincidan con el nombre de canal proporcionado, o puede buscar una coincidencia explícita. La coincidencia explícita le indica qué registro de autenticación de canal se utilizará si un canal intenta establecer una conexión desde una dirección IP específica, desde un gestor de colas específico o utilizando un ID de usuario específico y, opcionalmente, que declare un certificado personal SSL/TLS que contenga un nombre distinguido especificado.

Conceptos relacionados

"Seguridad de la mensajería remota" en la página 106

En este apartado se tratan aspectos relativos a la seguridad de la mensajería remota.

Interacción de CHLAUTH y CONNAUTH

Cómo interactúan los registros de autenticación de canal (CHLAUTH) y la autenticación de conexión (CONNAUTH) en IBM MQ, en el caso de una única conversación en un canal.

Distintos tipos de enlaces

IBM MQ admite dos métodos para una aplicación para conectarse:

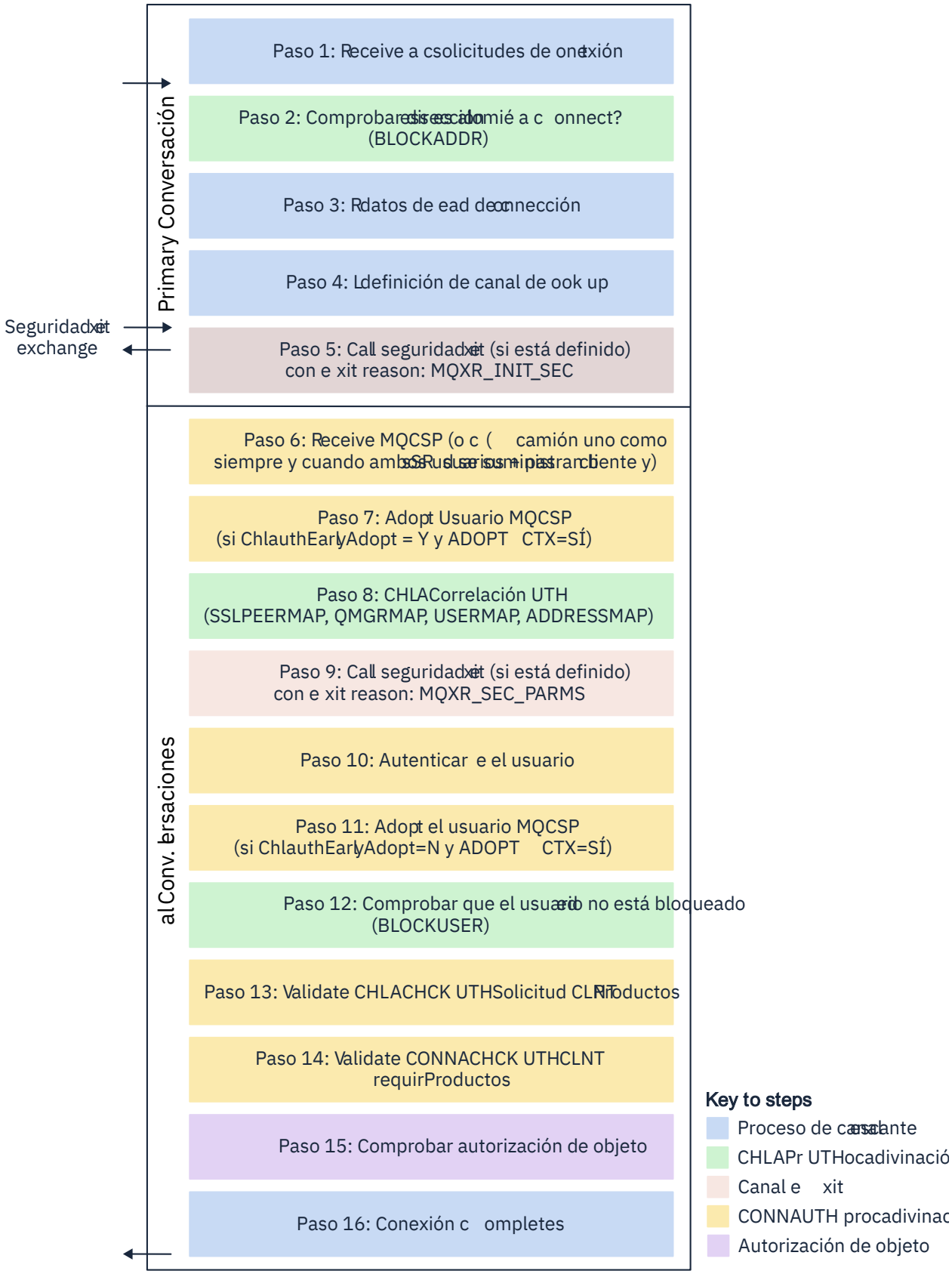
Enlaces locales
: Se aplica cuando la aplicación y el gestor de colas están en la misma imagen operativa. CHLAUTH no es relevante para este tipo de conexión de aplicación.

Enlaces de cliente
: Se aplica cuando la aplicación y el gestor de colas utilizan la red para comunicarse. La aplicación y el gestor de colas se pueden ejecutar en la misma máquina, o pueden estar en máquinas diferentes. En IBM MQ, una conexión de cliente se maneja en forma de un canal de conexión-servidor (SVRCONN) y, en esta situación, son aplicables ambos, CONNAUTH y CHLAUTH.

Pasos de enlaces del extremo de recepción de un canal

Cuando una aplicación se conecta a un gestor de colas, se realiza una cantidad considerable de comprobaciones para asegurarse de que ambos extremos del canal comprenden qué soporta el otro extremo. El extremo de recepción del canal realiza algunas comprobaciones adicionales, que implican CHLAUTH y CONNAUTH, para asegurarse de que el cliente está autorizado para conectarse, y este proceso también podría incluir una salida de seguridad, ya que esto puede afectar al resultado. También se ha referencia a esta fase de conexión del canal como la *fase de enlace*.

El diagrama siguiente lista los pasos que sigue un canal SVRCONN cuando se inicia el extremo del servidor (en el gestor de colas):



Paso 1: Recibir una solicitud de conexión
El iniciador de canal o el escucha recibe una solicitud de conexión de algún lugar de la red.

Paso 2: ¿La dirección está autorizada para conectarse?
Antes de que se pueda leer cualquier dato, IBM MQ comprueba la dirección IP del socio con respecto a las reglas CHLAUTH, para ver si la dirección está en la regla *BLOCKADDR*. Si la dirección no se encuentra y, por lo tanto, no está bloqueada, el flujo continúa hasta el siguiente paso.

Paso 3: Leer datos del canal
Ahora IBM MQ lee los datos en un almacenamiento intermedio y empieza a procesar la información enviada.

Paso 4: Buscar la definición de canal
En el primer flujo de datos, IBM MQ envía, entre otras cosas, el nombre del canal que el extremo emisor está intentando iniciar. El gestor de colas de recepción busca la definición de canal, que tiene todos los valores que se han especificado para el canal.

Paso 5: Llamar a la salida de seguridad (si hay alguna definida)
Si el canal tiene definida una salida de seguridad (SCYEXIT), se llama con la razón de salida (MQCXP.**ExitReason**) establecer en MQXR_INIT_SEC.

Paso 6: Recibir MQCSP
Si es necesario, construya uno si el cliente ha proporcionado las credenciales de autenticación.

Si el cliente es una aplicación Java o JMS que se ejecuta en modalidad de compatibilidad, el cliente no pasa una estructura MQCSP al gestor de colas. En su lugar, si la aplicación ha suministrado un ID de usuario y una contraseña, se genera aquí una estructura MQCSP.

Paso 7: Adopte el usuario MQCSP (si `ChlauthEarlyAdopt` es *Y* y ADOPTCTX=YES)
Las credenciales proporcionadas por el cliente se autentican.

Si CONNAUTH utiliza LDAP para correlacionar un nombre distinguido certificado con un ID de usuario corto, se produce la correlación en este paso.

Si la autenticación resulta satisfactoria, el canal adopta el ID de usuario y se utiliza en el paso de correlación CHLAUTH.

Nota: A partir de IBM MQ 9.0.4 se añade automáticamente el parámetro **`ChlauthEarlyAdopt=`** *Y* a la stanza de canales del archivo qm.ini para nuevos gestores de colas.

Paso 8: Correlación CHLAUTH
Se vuelve a inspeccionar la memoria caché CHLAUTH para buscar las reglas de correlación *SSLPEERMAP*, *USERMAP*, *QMGRMAP* y *ADDRESSMAP*.

Se utiliza la regla que coincide de forma más específica con el canal entrante. Si la regla tiene **USERSRC***(CHANNEL)* o *(MAP)*, el canal continúa en el enlace.

Si las reglas CHLAUTH se evalúan en una regla con **USERSRC***(NOACCESS)*, se bloquea la conexión de la aplicación con el canal, a menos que las credenciales se alteren temporalmente posteriormente con una credencial válida en el paso 9.

Paso 9: Llamar a la salida de seguridad (si hay alguna definida)
Si el canal tiene definida una salida de seguridad (SCYEXIT), se llama con la razón de salida (MQCXP.**ExitReason**) establecer en MQXR_SEC_PARMS.

Estará presente un puntero a MQCSP en el campo **SecurityParms** de la estructura MQCXP.

La estructura MQCSP tiene punteros al ID de usuario (MQCSP.**CSPUserIdPtr**) y contraseña (MQCSP.**CSPPasswordPtr**). V 9.4.0 A partir de IBM MQ 9.3.4, la estructura MQCSP también contiene un puntero a la señal de autenticación (MQCSP.**TokenPtr**).

Es posible cambiar el ID de usuario y la contraseña, y la señal de autenticación, en la salida. En el ejemplo siguiente se muestra cómo imprimiría una salida de seguridad los valores de ID de usuario y contraseña en un registro de auditoría:

```
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);
```

La salida puede indicar a IBM MQ que cierre el canal, devolviendo *MQXCC_CLOSE_CHANNEL* en MQCXP.Campo **Exitresponse** . De lo contrario, el proceso del canal continúa hasta la fase de autenticación de conexión.

Nota: Si la salida de seguridad cambia el usuario confirmado, las reglas de correlación CHLAUTH no se vuelven a aplicar al nuevo usuario.

Paso 10: Autenticar el usuario

La fase de autenticación se produce si CONNAUTH está habilitado en el gestor de colas.

Para comprobar esto, emita el mandato MQSC 'DISPLAY QMGR CONNAUTH'.

z/OS El ejemplo siguiente muestra la salida del mandato **DISPLAY QMGR CONNAUTH** de un gestor de colas que se ejecuta en 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 El ejemplo siguiente muestra la salida del mandato '**DISPLAY QMGR CONNAUTH**' de un gestor de colas que se ejecuta en IBM MQ for Multiplatforms.

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

El valor de CONNAUTH es el nombre de un objeto **AUTHINFO** IBM MQ.

Puesto que la autenticación del sistema operativo (**AUTHTYPE**(*IDPWOS*)) es válida en ambos sistemas, IBM MQ for Multiplatforms y IBM MQ for z/OS, el ejemplo utiliza la autenticación del sistema operativo.

z/OS El ejemplo siguiente muestra el objeto AUTHINFO predeterminado con **AUTHTYPE**(*IDPWOS*) desde un gestor de colas que se ejecuta en 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 El ejemplo siguiente muestra el objeto AUTHINFO predeterminado con **AUTHTYPE**(*IDPWOS*) desde un gestor de colas que se ejecuta en 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)
```

El objeto AUTHINFO TYPE (IDPWOS) tiene un atributo denominado CHCKCLNT. Si el valor se cambia a *REQUIRED* , todas las aplicaciones cliente tienen que proporcionar credenciales válidas.

Si el usuario se ha autenticado en el paso 7, no se realizará otra comprobación de autenticación a menos que:

- El ID de usuario y la contraseña, o la señal de autenticación, en el campo SecurityParms de la estructura MQCXP se ha cambiado mediante una salida de seguridad en el paso 9.
- La aplicación cliente se ha conectado con opciones que solicitan funcionalidad reconectable.

Paso 11: Adoptar el contexto del usuario MQCSP (Si ChlauthEarlyAdopt=*N* y ADOPTCTX=YES)
Puede establecer el atributo ADOPTCTX, que controla si el canal se ejecuta bajo MCAUSER, o bajo el ID de usuario que ha proporcionado la aplicación.

Si el ID de usuario confirmado en el campo MQCSP, o **SecurityParms** de la estructura MQXCP, se ha autenticado correctamente y **ADOPTCTX** es *YES*, el contexto del usuario resultante de los pasos 7 y 8 se adopta como el contexto a utilizar para esta aplicación, a menos que el ID de usuario y la contraseña, o la señal de autenticación, en el campo **SecurityParms** de la estructura MQCXP se haya modificado mediante una salida de seguridad en el paso 9.

El ID de usuario certificado es el que se ha comprobado la autorización para utilizar los recursos de IBM MQ.

Por ejemplo, no tiene un MCAUSER establecido en el canal SVRCONN y el cliente se está ejecutando bajo '`johndoe`' en la máquina Linux. La aplicación especifica el usuario '`fred`' en MQCSP, así que el canal inicia la ejecución con '`johndoe`' como el MCAUSER activo. Después de la comprobación de CONNAUTH, se adopta el usuario '`fred`' y el canal se ejecuta con '`fred`' como el MCAUSER activo.

Paso 12: Comprobar que el usuario no esté bloqueado (BLOCKUSER)
Si la comprobación CONNAUTH es satisfactoria, la memoria caché CHLAUTH se inspecciona de nuevo para comprobar si el MCAUSER activo está bloqueado por una regla *BLOCKUSER* . Si el usuario está bloqueado, el canal finaliza.

Paso 13: Validar requisitos CHLAUTH CHCKCLNT
Si la regla CHLAUTH que se ha seleccionado en el paso 8 especifica adicionalmente un valor CHCKCLNT de REQUIRED o REQDADM, se realiza la validación para asegurarse de que se ha proporcionado un ID de usuario CONNAUTH válido para cumplir el requisito.

- Si se establece CHCKCLNT (REQUIRED), un usuario debe haberse autenticado en el paso 7 o 10. De lo contrario, se rechazará la conexión.
- Si se establece CHCKCLNT (REQDADM), un usuario debe haberse autenticado en el paso 7 o 10 si se determina que esta conexión es privilegiada. De lo contrario, se rechazará la conexión.
- Si se establece CHCKCLNT (ASQMGR), se omite este paso.

Notas:

1. Si se establece CHCKCLNT (REQUIRED) o CHCKCLNT (REQDADM), pero CONNAUTH no está habilitado en el gestor de colas, la conexión falla con un código de retorno MQRC_SECURITY_ERROR (2063) debido al conflicto en la configuración.
2. El usuario no se vuelve a autenticar en este paso.

Paso 14: Validar requisitos CONNAUTH CHCKCLNT.
La fase de autenticación se produce si CONNAUTH está habilitado en el gestor de colas.

Se comprueba el valor CONNAUTH CHCKCLNT para determinar qué requisitos se establecen para las conexiones entrantes:

- Si se establece CHCKCLNT (NONE), se omite este paso
- Si se establece CHCKCLNT (OPTIONAL), este paso se omite.
- Si se establece CHCKCLNT (REQUIRED), un usuario debe haberse autenticado en el paso 7 o 10. De lo contrario, se rechazará la conexión.

- Si se establece CHCKCLNT (REQDADM), un usuario debe haberse autenticado en el paso 7 o 10 si se determina que esta conexión es privilegiada. De lo contrario, se rechazará la conexión.

Nota: El usuario no se vuelve a autenticar en este paso.

Multi **Paso 15: Comprobar autorización de objeto**
Se realiza una comprobación para asegurarse de que el MCAUSER activo tiene la autorización adecuada para conectarse al gestor de colas.

ALW Consulte Gestor de autorizaciones sobre objetos, si desea más información.

IBM i Consulte “Gestor de autorizaciones sobre objetos (OAM) en IBM i” en la página 165 para obtener más información.

Paso 16: La conexión se completa
Si los pasos precedentes se completan satisfactoriamente, la conexión se completa.

Conceptos relacionados
CONNAUTH
Un gestor de colas se puede configurar para autenticar las credenciales proporcionadas por una aplicación cuando se conecta.

Referencia relacionada
SET CHLAUTH
ALTER AUTHINFO

Resolución de problemas de acceso de CHLAUTH
Pasos y ejemplos para resolver determinados problemas de acceso al utilizar registros de autenticación de canal (CHLAUTH).

Antes de empezar

Nota: Los pasos de esta tarea requieren que ejecute mandatos MQSC. La forma de hacer esto varía según la plataforma. Ver AdministrarIBM MQ usando comandos MQSC .

Acerca de esta tarea

Existen tres reglas predeterminadas para el proceso de CHLAUTH:

- NO ACCESS (sin acceso) en todos los canales por parte de usuarios `MQ-admin*`
- NO ACCESO a todos los SYSTEM.* canales por todos los usuarios
- ALLOW (permitir) acceso al canal SYSTEM.ADMIN.SVRCONN (usuarios no `MQ-admin`)

Las primeras dos reglas bloquean el acceso a todos los canales. La tercera regla es más específica y, por lo tanto, tiene prioridad sobre las otras dos, si el canal es el canal SYSTEM.ADMIN.SVRCONN, permitiendo así el acceso a dicho canal.

Las reglas CHLAUTH se utilizan para determinar si un canal se puede iniciar, y permiten la correlación, a través de MCAUSER con otro ID de usuario. Si el canal no se puede iniciar, normalmente se producen los errores siguientes:

- `RC 2035 MQRC_NOT_AUTHORIZED`
- `RC 2059 MQRC_Q_MGR_NOT_AVAILABLE`
- `AMQ4036 Acceso no permitido`
- `AMQ9776: El canal estaba bloqueado por el ID de usuario`
- `AMQ9777: El canal estaba bloqueado`
- `MQJE001: Se ha producido una MQException. Código de terminación 2, Razón 2035`
- `MQJE036: El gestor de colas ha rechazado un intento de conexión`

Debería bloquear el acceso de forma estricta y, después, añadir más reglas CHLAUTH para controlar quién puede acceder e iniciar reglas.

Como medida temporal, y para resolver los errores listados, realice cualquiera de los pasos siguientes.

Procedimiento

- **Inhabilitar reglas CHLAUTH**

  Como medida temporal y, también, para resolver los errores anteriores, puede inhabilitar reglas CHLAUTH. Las reglas se pueden volver a habilitar en cualquier momento, y si la inhabilitación de las reglas CHLAUTH resuelve el problema de conexión, sabe que esta fue la causa.

  Para inhabilitar las reglas CHLAUTH, ejecute el siguiente mandato MQSC:

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

  Tenga en cuenta que también puede establecer CHLAUTH en *WARN*, que permite acceder a y registrar el resultado de la regla.

- **Modificar o eliminar reglas CHLAUTH**

  También puede suprimir o modificar la regla o reglas CHLAUTH que están provocando el problema.

  Para modificar una regla CHLAUTH, utilice el mandato SET CHLAUTH con ACTION (REPLACE). Por ejemplo, para modificar la regla predeterminada que hace que ningún usuario de `MQ-admin` acceda a todos los canales a WARN, en lugar de estar bloqueado, ejecute el siguiente mandato MQSC:

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

  Para suprimir una regla CHLAUTH, utilice el mandato SET CHLAUTH con ACTION (REMOVE). Por ejemplo, para suprimir la regla predeterminada que no hace que ningún usuario de `MQ-admin` acceda a todos los canales, ejecute el siguiente mandato MQSC:

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

- **Probar acceso utilizando MATCH (RUNCHECK)**

  Puede probar el resultado de las reglas CHLAUTH, utilizando la opción MATCH (*RUNCHECK*) de la regla CHLAUTH. La opción **MATCH** (*RUNCHECK*) devuelve el registro que ha comparado un canal de entrada específico durante el tiempo de ejecución, si dicho canal se conecta en este gestor de colas. Debe proporcionar:

  – El nombre del canal
  – Atributo ADDRESS
  – Atributo SSLPEER, solo si el canal de entrada utiliza SSL o TLS
  – QMNAME, si el canal de entrada es un canal de gestor de colas, o
  – Atributo CLNTUSER, si el canal de entrada es un canal cliente

  El ejemplo siguiente ejecuta un mandato MQSC para comprobar qué regla CHLAUTH, con las reglas predeterminadas en vigor, da como resultado que un `MQ-admin` usuario `johndoe` acceda a un 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 el usuario `johndoe`, el canal no se ejecuta, el usuario se bloqueará debido a la regla BLOCKUSER para usuarios *MQADMIN.

El ejemplo siguiente ejecuta un mandato MQSC para comprobar qué regla CHLAUTH, con las reglas predeterminadas en su lugar, da como resultado el usuario `alice` que no es un usuario de `MQ-admin` , que accede a un canal denominado `CHAN1`:

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

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

Para el usuario `alice`, el canal se ejecuta y el canal pasa `alice` como el MCAUSER. El MCAUSER es el ID de usuario utilizado para comprobar las autoridades sobre objeto IBM MQ.

Referencia relacionada
SET CHLAUTH
DISPLAYCHLAUTH

Creación de nuevas reglas CHLAUTH para usuarios
Algunos escenarios comunes para los usuarios y reglas CHLAUTH de ejemplo para lograrlos.

Antes de empezar

Nota: Los pasos de esta tarea requieren que ejecute mandatos MQSC. La forma de hacer esto varía según la plataforma. Ver AdministrarIBM MQ usando comandos MQSC .

Acerca de esta tarea

Existen tres reglas predeterminadas para el proceso de CHLAUTH:

- NO ACCESS (sin acceso) en todos los canales por parte de usuarios `MQ-admin*`
- NO ACCESO a todos los SYSTEM.* canales por todos los usuarios
- ALLOW (permitir) acceso al canal SYSTEM.ADMIN.SVRCONN (usuarios no `MQ-admin`)

Las primeras dos reglas bloquean el acceso a todos los canales. La tercera regla es más específica y, por lo tanto, tiene prioridad sobre las otras dos, si el canal es el canal SYSTEM.ADMIN.SVRCONN, permitiendo así el acceso a dicho canal.

Para crear nuevas reglas CHLAUTH para los usuarios, configure uno o varios de los escenarios siguientes.

Procedimiento

- **Control de acceso para usuarios específicos de `MQ-admin`**
  a) Configure un canal de conexión de servidor que se utilizará exclusivamente para una perspectiva administrativa, es decir, para conectarse desde IBM MQ Explorer.

     Tiene un canal específico para este uso y la dirección o direcciones IP definidas, desde las cuales desea que se acepten conexiones, y el acceso bloqueado para el ID `'mqm'`, si la conexión no procede de una de las direcciones IP especificadas.
  b) Realice un canal SVRCONN para los usuarios de IBM MQ Explorer y `MQ-admin` denominado `ADMIN.CHAN`.

     Ejecute el siguiente mandato MQSC:

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

  c) Para la realización de pruebas, asegúrese de que tiene un usuario definido que esté en el grupo `MQ-admin` y un usuario que no lo esté.

     Para este escenario, `mqadm` está en el grupo `MQ-admin` y `alice` no lo está.

d) Confirme que las reglas CHLAUTH predeterminadas están en su lugar.

e) Añada tres reglas para permitir que un usuario específico acceda a ADMIN.CHAN como `MQ-admin` desde determinadas direcciones IP:

- Establecer NOACCESS desde cualquier dirección
- Establecer BLOCKUSER para este canal para solo bloquear el usuario nobody, que altera temporalmente el *MQADMIN BLOCKUSER
- ALLOW acceso al usuario `mqadm` en una subred específica de direcciones y MAP con la autoridad de usuario `mqadm`

Para ello, ejecute los siguientes mandatos MQSC:

```
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)
```

En este punto, el usuario `mqadm` puede acceder e iniciar el canal ADMIN.CHAN, desde el rango de direcciones IP especificado.

f) Opcional: Puede ejecutar el mandato MQSC MATCH (RUNCHECK) en cualquier momento para ver los resultados de cada uno de estos mandatos:

```
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)
```

En este punto, solo los usuarios que tienen un registro CHLAUTH están autorizados para acceder al uso de ADMIN.CHAN.

- **Control de acceso para un usuario específico y una aplicación cliente de IBM MQ**

  Para este escenario, las reglas CHLAUTH predeterminadas son adecuadas, suponiendo que se debe establecer la autorización IBM MQ para un usuario específico, para proporcionar la autorización IBM MQ correcta (utilizando setmqaut).

  En este escenario, las autorizaciones se establecen para un usuario `mqapp1`, que no es un usuario `MQ-admin`.

  a) Utilice el siguiente mandato MQSC para crear un canal SVRCONN, APP1.CHAN, que utilizará una aplicación determinada y un usuario específico.

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

b) Con las reglas CHLAUTH predeterminadas en vigor, el usuario `mqapp1` puede iniciar el canal APP1.CHAN.

El ID de usuario que procede de la aplicación cliente IBM MQ se utiliza para la comprobación de autoridades de objeto IBM MQ. En este caso, suponiendo que el usuario `mqapp1` esté ejecutando la aplicación cliente IBM MQ , se utiliza para la comprobación de autorización sobre objetos de IBM

MQ . Por lo tanto, si mqapp1 tiene acceso a los objetos IBM MQ que necesita la aplicación, todo está correcto; en caso contrario, obtendrá errores de autorización.

Puede aumentar más la seguridad creando reglas CHLAUTH específicas para el ID de usuario mqapp1, pero bajo las reglas predeterminadas, ningún miembro del grupo MQ-admin puede acceder a este canal.

Ejecute los siguientes mandatos MQSC:

```
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)
```

- **Controlar el acceso de un usuario específico utilizando el nombre distinguido (DN) de certificado de dicho usuario**

  Para este escenario, el usuario debe tener un certificado que se haya trasladado al gestor de colas. El DN se compara con el valor SSLPEER de la regla CHLAUTH, y SSLPEER puede utilizar caracteres comodín.

  Si coinciden, el usuario también se puede correlacionar con un MCAUSER diferente para finalidades de comprobación de las autoridades sobre objetos IBM MQ. La correlación de MCAUSER puede minimizar el número de usuarios que se deben gestionar en el gestor de autorizaciones sobre objetos (OAM) de IBM MQ.

  a) Tiene un canal TLS con certificados en uso, y necesita reglas para:

  – Bloquear a todos los usuarios para un canal determinado
  – Permitir solo a los usuarios con un SSLPEER concreto que utilizan el cliente de dicho usuario para el acceso de OAM IBM MQ.

  Ejecute los siguientes mandatos MQSC:

```
.
# 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)
```

  El ID de usuario del cliente que se conecta en el canal se utiliza para la autoridad de OAM IBM MQ de objetos IBM MQ; por lo tanto, el ID de usuario debe tener las autoridades de IBM MQ apropiadas.

  b) Opcional: Correlacione con un ID de usuario de IBM MQ diferente.

  Vuelva a ejecutar el mandato MQSC anterior, sustituyendo USERSRC(MAP) MCAUSER('mquser1') por USERSRC(CHANNEL).

- **Correlacionar un usuario determinado con el usuario mqm**

  Se trata de una adición o modificación a Control de acceso para usuarios específicos de MQ-admin.

  Utilice los mandatos MQSC para añadir la siguiente regla CHLAUTH para correlacionar usuarios concretos con el usuario mqm , o un ID de usuario MQ-admin , que tenga la autorización de objeto IBM MQ configurada en el OAM de 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)
```

Esto habilita y correlaciona el usuario `johndoe` a través del usuario `mqm` para el canal ADMIN.CHAN concreto.

Conceptos relacionados

“Creación de nuevas reglas CHLAUTH para canales” en la página 70
Para ayudarle a crear sus propias reglas CHLAUTH, aquí hay algunos casos de ejemplo comunes para los canales y reglas CHLAUTH de ejemplo para llevar a cabo esto.

Tareas relacionadas

“Resolución de problemas de acceso de CHLAUTH” en la página 65
Pasos y ejemplos para resolver determinados problemas de acceso al utilizar registros de autenticación de canal (CHLAUTH).

Referencia relacionada

SET CHLAUTH

DISPLAYCHLAUTH

Creación de nuevas reglas CHLAUTH para canales

Para ayudarle a crear sus propias reglas CHLAUTH, aquí hay algunos casos de ejemplo comunes para los canales y reglas CHLAUTH de ejemplo para llevar a cabo esto.

En este tema se incluyen los escenarios siguientes:

- “Solo permitir acceso a un canal concreto desde un rango de direcciones IP específicas.” en la página 70
- “Para un canal específico, bloquear a todos los usuarios, pero permitir la conexión de usuarios específicos.” en la página 70
- “Utilización de CHLAUTH para canales receptor y emisor” en la página 71

Solo permitir acceso a un canal concreto desde un rango de direcciones IP específicas.

Para este escenario desea:

- Establecer Sin acceso en el canal desde cualquier lugar
- Permitir el acceso desde un rango de dirección o direcciones IP específicas

```
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)
```

Esto permite que solo se inicie el canal APP2.CHAN cuando la conexión procede del rango de direcciones IP específicas especificado.

El usuario que se conecta como MCAUSER se correlaciona con `mqapp2` y, por lo tanto, obtiene la autorización del OAM de IBM MQ para dicho usuario.

Para un canal específico, bloquear a todos los usuarios, pero permitir la conexión de usuarios específicos.

Existen tres reglas predeterminadas para el proceso de CHLAUTH:

- NO ACCESS (sin acceso) en todos los canales por parte de usuarios `MQ-admin*`
- NO ACCESO a todos los SYSTEM.* canales por todos los usuarios
- ALLOW (permitir) acceso al canal SYSTEM.ADMIN.SVRCONN (usuarios no `MQ-admin`)

Las primeras dos reglas bloquean el acceso a todos los canales. La tercera regla es más específica y, por lo tanto, tiene prioridad sobre las otras dos, si el canal es el canal SYSTEM.ADMIN.SVRCONN, permitiendo así el acceso a dicho canal.

Para este escenario, el acceso al canal MY.SVRCONN tiene las reglas CHLAUTH predeterminadas.

Tendrá que añadir lo siguiente:

```
# 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)
```

La primera parte del código bloquea a cualquier usuario de la conexión en MY.SVRCONN, a continuación, el código solo permite que se inicie el canal MY.SVRCONN cuando la conexión procede del ID de usuario específico `johndoe`.

El usuario que se conecta en el canal `johndoe` se utiliza para la autoridad de OAM IBM MQ de los objetos IBM MQ. Por lo tanto, el ID de usuario debe tener las autorizaciones de IBM MQ apropiadas.

Puede correlacionarse con un ID de usuario de IBM MQ diferente, si lo desea, utilizando:

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

en lugar de USERSRC(CHANNEL).

Utilización de CHLAUTH para canales receptor y emisor

Puede utilizar reglas CHLAUTH para añadir una seguridad adicional a los canales receptor y emisor, para restringir el acceso al canal receptor. Tenga en cuenta que, si está añadiendo o realizando cambios en reglas CHLAUTH, las reglas CHLAUTH actualizadas solo se aplican al iniciar el canal, de forma que si los canales ya se están ejecutando, tendrá que detenerlos y reiniciarlos para que se apliquen las actualizaciones de CHLAUTH.

Las reglas CHLAUTH se pueden utilizar en cualquier canal, pero existen algunas restricciones. Por ejemplo, las reglas USERMAP se aplican solo a canales SVRCONN.

Este ejemplo permite una conexión desde una dirección IP concreta solo para iniciar el 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 ejemplo solo permite la conexión desde un gestor de colas concreto:

```
# 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)
```

Tareas relacionadas

“Resolución de problemas de acceso de CHLAUTH” en la página 65

Pasos y ejemplos para resolver determinados problemas de acceso al utilizar registros de autenticación de canal (CHLAUTH).

"Creación de nuevas reglas CHLAUTH para usuarios" en la página 67
Algunos escenarios comunes para los usuarios y reglas CHLAUTH de ejemplo para lograrlos.

Referencia relacionada
SET CHLAUTH
DISPLAYCHLAUTH

Creación de una regla de detención posterior CHLAUTH
Al pensar en el control de las conexiones de entrada en el gestor de colas, tiene dos opciones.
Puede intentar listar todas las conexiones que no están permitidas, o puede empezar diciendo que no están permitidas todas las conexiones y, a continuación, intentar listar todas las conexiones que están permitidas. Esta segunda opción se describe aquí.

Acerca de esta tarea

La razón por la que se utiliza la segunda opción es que, si intenta listar todas las conexiones que no están permitidas y, por lo tanto, todo lo que no está en la lista está permitido, el resultado de que falta una de la lista es que una conexión que no debería estar permitida es capaz de conectarse, lo que provoca una posible infracción de seguridad.

Por el contrario, si en su lugar, empieza diciendo que no se permite cada conexión y, a continuación, lista las que están, el resultado de que falte una de esta lista no es una infracción de seguridad. Si su empresa requiere que se añadan conexiones adicionales, se trata de una tarea relativamente sencilla, pero no existe una posible brecha de seguridad.

Lo primero que hay que hacer es crear una regla *back-stop* , que es una regla que captura las conexiones que de otro modo no coincidían con reglas más específicas. Esta regla tiene el efecto de impedir que las conexiones remotas se puedan conectar al gestor de colas en absoluto.

Sin embargo, si le preocupa este enfoque, puede configurar la regla *back-stop* en modalidad de aviso; consulte el paso "2" en la página 72

Procedimiento

1. Para crear una regla de detención posterior que detenga las conexiones remotas que se conectan al gestor de colas, emita el mandato siguiente:

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

   Ahora que ha cerrado la puerta en todas las conexiones remotas, puede empezar a aplicar reglas más específicas para permitir la entrada de determinadas conexiones. Por ejemplo:

   ```
   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. Si desea crear la regla de detención posterior en modalidad de aviso, emita el mandato siguiente:

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

   Ahora puedes continuar, y hacer todas tus reglas positivas. Cuando crea que ha creado todas las reglas que necesita, active los sucesos de canal emitiendo el mandato siguiente:

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

y supervise el SYSTEM.ADMIN.CHANNEL.EVENT para sucesos con **Reason** establecido en MQRC_CHANNEL_BLOCKED_WARNING.

Estos sucesos detallan las conexiones que han coincidido con la regla de detención posterior, pero debido a que el mandato se está ejecutando en modalidad de aviso, no se han bloqueado realmente por el momento.

Revise cada uno de estos sucesos y determine si esta conexión debe tener una regla positiva para permitirla, o si se ha comparado correctamente con la regla *back-stop* . Puede ejecutar en esta modalidad, revisando los sucesos a medida que se crean, hasta que esté satisfecho de que haya visto todos los canales de entrada y tenga las reglas positivas adecuadas para todos ellos.

En este punto, puede cambiar la regla *back-stop* para empezar a bloquear realmente las conexiones que coincidan emitiendo el mandato siguiente:

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

Creación de un administrador de IBM MQ sin privilegios
Cómo crear un administrador de IBM MQ sin privilegios utilizando CHLAUTH.

Acerca de esta tarea

En el contexto de esta tarea, los términos:

Uso privilegiado
Significa un usuario que tiene autorización para realizar una operación sin que se le otorgue explícitamente acceso para realizar dicha operación. Los usuarios del grupo mqm son ejemplos de estos usuarios privilegiados.

Administrador de IBM MQ
Es un usuario que tiene la necesidad de emitir mandatos administrativos para IBM MQ, como por ejemplo **DEFINE QLOCAL** o **START CHANNEL**.

Los pasos siguientes crean un administrador de IBM MQ sin privilegios.

Procedimiento

1. Cree un ID de usuario en la máquina del gestor de colas utilizando los mandatos adecuados para la plataforma o plataformas que utiliza la empresa.

   En este ejemplo se utiliza el nombre de usuario `alice` .
2. Otorgue a este nuevo usuario autorización para emitir todos los mandatos administrativos de IBM MQ llevando a cabo el procedimiento siguiente:
   a) Inicie IBM MQ Explorer utilizando un usuario privilegiado.
   b) Vaya al *Asistente basado en roles* seleccionando el gestor de colas adecuado y, a continuación, `Autorizaciones de objeto` y `Añadir autorizaciones basadas en roles`.
   c) En el panel del asistente que aparece, especifique el ID de usuario que ha creado en el primer paso, o si prefiere trabajar con grupos, especifique el nombre de grupo para el usuario o conjunto de usuarios que desea convertir en administradores de IBM MQ no privilegiados.
   d) Configure el asistente para el acceso administrativo completo.
   e) Si desea permitir que el administrador de IBM MQ no privilegiado pueda examinar mensajes en colas, seleccione también ese recuadro de selección.
   f) Revise los mandatos en el panel de vista previa en la parte inferior del asistente.

      Puede cortar y pegar estos mandatos para crear sus propios scripts.

      Una razón por la que puede preferir hacer esto con su propio script es reducir la cantidad de acceso que da a este usuario. Quizás en lugar de otorgar acceso a todos los objetos, es posible que prefiera sólo otorgar acceso a un determinado grupo de objetos.

Al pulsar **Aceptar** en el asistente se emiten los mandatos tal como se muestran.

g) Debe configurar algunas reglas CHLAUTH para permitir el acceso remoto para este ID de usuario, si el requisito para un administrador de IBM MQ no privilegiado es que sea también para el acceso remoto.

Suponiendo que la empresa esté utilizando la guía de "Creación de una regla de detención posterior CHLAUTH" en la página 72, todo lo que debe hacer es añadir una regla de habilitación.

La regla que cree más bien depende de cómo elija autenticar los administradores remotos de IBM MQ .

Si utiliza una autenticación TCP/IP débil, puede configurar una regla CHLAUTH similar a la siguiente:

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

9. Si utiliza la autenticación TLS, puede configurar una regla CHLAUTH similar a la siguiente:

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

Ahora, cuando un usuario se conecta a `admin-channel-name` (y coincide con las reglas CHLAUTH), puede emitir mandatos con el ID de usuario `alice` en el gestor de colas y, por lo tanto, no es necesario el acceso remoto privilegiado.

Autenticación de conexión

La autenticación de conexión permite a las aplicaciones proporcionar credenciales de autenticación cuando se conectan a un gestor de colas. El gestor de colas valida las credenciales. El ID de usuario proporcionado en las credenciales también se puede adoptar para su uso en comprobaciones de autorización para los recursos a los que accede la aplicación.

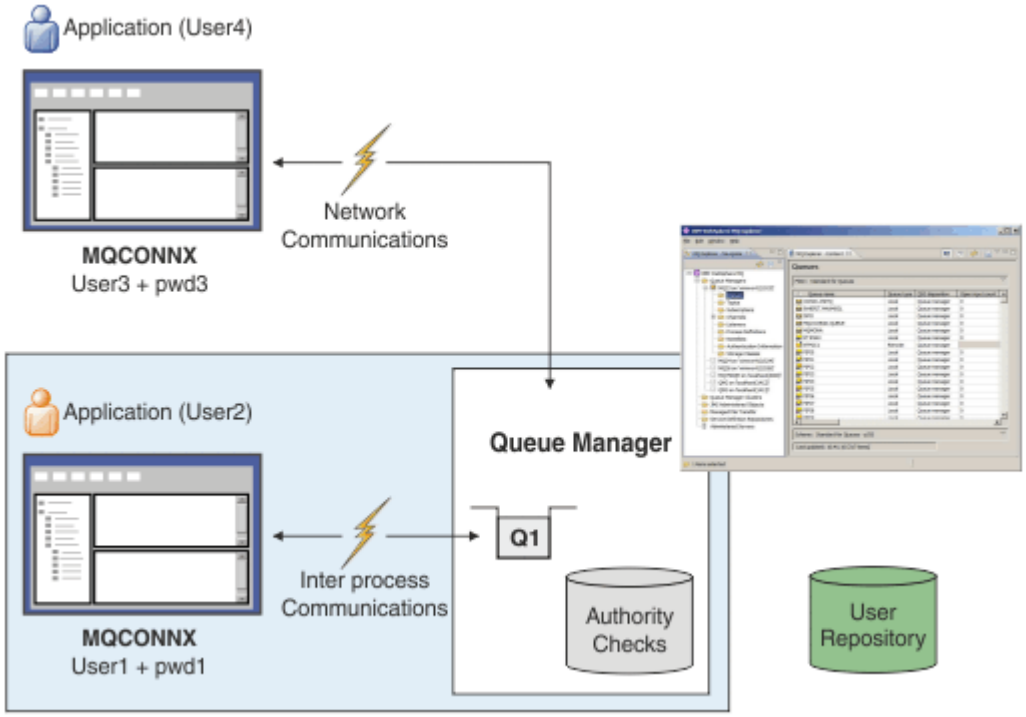
Las aplicaciones pueden proporcionar un ID de usuario y una contraseña para la autenticación cuando se conectan a un gestor de colas.

V 9.4.0 A partir de IBM MQ 9.3.4, las aplicaciones IBM MQ client también pueden proporcionar una señal de autenticación como método alternativo de autenticación.

El gestor de colas se puede configurar para validar las credenciales proporcionadas por la aplicación.

Un ID de usuario y una contraseña proporcionados por una aplicación se comprueban utilizando el repositorio de usuarios en la configuración del gestor de colas. Para obtener más información sobre el repositorio que se utiliza para comprobar los ID de usuario y las contraseñas, consulte Repositorios de usuarios.

V 9.4.0 Las señales de autenticación se validan utilizando los certificados y las claves simétricas en el almacén de claves de autenticación de señales del gestor de colas para validar la firma de la señal. Para obtener más información sobre la autenticación de usuarios con señales de autenticación, consulte "Cómo trabajar con señales de autenticación" en la página 331.



En el diagrama, dos aplicaciones están realizando conexiones con un gestor de colas, una aplicación es una aplicación cliente y la otra utiliza enlaces locales. Las aplicaciones pueden utilizar diversas API para conectarse al gestor de colas, pero todas tienen la capacidad de proporcionar un ID de usuario y una contraseña. El ID de usuario bajo el cual se está ejecutando la aplicación, `User2` y `User4` en el diagrama, que es el ID de usuario del sistema habitual presentado en IBM MQ, podría ser diferente del ID de usuario proporcionado por la aplicación, `User1` y `User3`.

El gestor de colas recibe mandatos de configuración (en el diagrama, se utiliza IBM MQ Explorer) y gestiona la apertura de recursos y comprueba la autorización para acceder a esos recursos. Existen muchos recursos diferentes en IBM MQ para los que una aplicación podría solicitar autorización de acceso. El diagrama ilustra la apertura de una cola de salida, pero se aplican los mismos principios a otros recursos también.

Conceptos relacionados

“Autenticación de conexión: Configuración” en la página 75
Un gestor de colas se puede configurar para autenticar las credenciales proporcionadas por una aplicación cuando se conecta.

“Autenticación de conexión: Cambios en la aplicación” en la página 80

“Autenticación de conexión: Depósitos de usuario” en la página 81
Para cada uno de los gestores de colas, puede seleccionar diferentes tipos de objetos de información de autenticación para autenticar los ID de usuario y las contraseñas.

Autenticación de conexión: Configuración

Un gestor de colas se puede configurar para autenticar las credenciales proporcionadas por una aplicación cuando se conecta.

Activación de la autenticación de la conexión en un gestor de colas

En un objeto de gestor de colas, el atributo **CONNAUTH** puede establecerse en el nombre de un objeto de información de autenticación (AUTHINFO). El atributo **AUTHTYPE** de un objeto AUTHINFO especifica el tipo del objeto. Los objetos AUTHINFO que se utilizan para la autenticación de conexión pueden ser de uno de los dos tipos siguientes:

IDPWOS
El gestor de colas utiliza el sistema operativo local para autenticar el ID de usuario y la contraseña proporcionados por una aplicación de conexión.

A partir de IBM MQ 9.3.4, este tipo de objeto AUTHINFO también permite a un gestor de colas que se ejecuta en AIX o Linux validar señales de autenticación. Además del objeto AUTHINFO que se utiliza para configurar la autenticación de conexión, el gestor de colas debe estar configurado para aceptar señales de autenticación con la stanza **AuthInfo** del archivo `qm.ini` . Para obtener más información sobre cómo configurar un gestor de colas para aceptar señales de autenticación, consulte “Configuración de un gestor de colas para aceptar señales de autenticación utilizando un almacén de claves local” en la página 338.

IDPWLDAP
El gestor de colas utiliza un servidor LDAP para autenticar el ID de usuario y la contraseña proporcionados por una aplicación de conexión.

Nota: No puede especificar ningún otro tipo de objeto de información de autenticación en el atributo **CONNAUTH** del gestor de colas.

Los objetos AUTHINFO de tipo `IDPWOS` e `IDPWLDAP` son similares en varios de sus atributos. Los atributos descritos aquí son comunes a ambos tipos de objetos.

Los siguientes mandatos MQSC de ejemplo activan la autenticación de conexión con las operaciones siguientes:

1. Defina un objeto AUTHINFO denominado `USE.PW`.
2. Modifique el atributo **CONNAUTH** del gestor de colas para que haga referencia a este objeto AUTHINFO.
3. Emita el mandato **REFRESH SECURITY** para renovar la configuración de autenticación de conexión del gestor de colas. El mandato **REFRESH SECURITY** debe emitirse antes de que el gestor de colas reconozca los cambios en la configuración de autenticación de conexión.

```
DEFINE AUTHINFO(USE.PW) +
AUTHTYPE(IDPWOS) +
FAILDLAY(10) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED)

ALTER QMGR CONNAUTH(USE.PW)

REFRESH SECURITY TYPE(CONNAUTH)
```

Para controlar si se comprueban las credenciales para las conexiones realizadas por aplicaciones enlazadas localmente, utilice el atributo AUTHINFO **CHCKLOCL** (comprobar conexiones locales). Para controlar si se comprueban las credenciales para las conexiones realizadas por las aplicaciones cliente, utilice el atributo AUTHINFO **CHCKCLNT** (comprobar conexiones de cliente).

CHCKLOCL acepta los valores de `NONE` y `OPTIONAL`, y **CHCKCLNT** permite que se configure el valor de `NONE` para los requisitos de autenticación:

NONE
Las credenciales de autenticación proporcionadas por las aplicaciones no se comprueban.

OPTIONAL
Garantiza que las credenciales proporcionadas por una aplicación son válidas. Sin embargo, no es obligatorio que las aplicaciones proporcionen credenciales de autenticación. Esta opción puede resultar de utilidad durante la migración, por ejemplo.

Si:

- Proporcione el nombre de usuario y la contraseña, se autentican.
- No proporcione el nombre de usuario y la contraseña, se permite la conexión.
- Proporcione el nombre de usuario, pero no la contraseña que recibe un error.

Importante: `OPCIONAL` es el valor mínimo que puede establecer si también desea establecer una opción más restrictiva en las reglas de autenticación de canal (CHLAUTH).

Si selecciona NONE y la conexión de cliente coincide con un registro CHLAUTH con **CHCKCLNT** establecido en REQUIRED (o REQDADM en plataformas distintas de z/OS), la conexión falla. Recibe el mensaje AMQ9793 en Multiplatforms y el mensaje CSQX793E en z/OS.

Para obtener más información sobre el uso de reglas de autenticación de canal para establecer opciones **CHCKCLNT** más restrictivas para algunas conexiones de cliente, consulte "Granularidad de la configuración" en la página 77.

REQUIRED
Requiere que todas las aplicaciones proporcionen credenciales válidas. Consulte también la nota siguiente.

REQDADM
Los usuarios privilegiados deben proporcionar credenciales válidas, pero los usuarios no privilegiados se tratan como con el valor OPTIONAL . Consulte también la nota siguiente. z/OS (Este valor no está permitido en sistemas z/OS).

Nota:

Si se establece **CHCKLOCL** en REQUIRED o REQDADM significa que no puede administrar localmente el gestor de colas utilizando **runmqsc** (error AMQ8135: No autorizado) a menos que el usuario especifique el parámetro **-u** para especificar el ID de usuario en el mandato **runmqsc** . Con ese parámetro establecido, **runmqsc** solicita la contraseña del usuario en la consola.

De forma similar, un usuario que ejecuta IBM MQ Explorer en el sistema local verá el error AMQ4036 al intentar conectarse al gestor de colas. Para especificar un ID de usuario y una contraseña, pulse con el botón derecho del ratón en el objeto del gestor de colas local y seleccione **Detalles de conexión** > **Propiedades ...** en el menú. En la sección **ID de usuario** , especifique el ID de usuario y la contraseña que se van a utilizar y, a continuación, pulse **Aceptar**.

Se aplican consideraciones similares a las conexiones remotas con **CHCKCLNT**.

El atributo **CONNAUTH** del gestor de colas está en blanco para los gestores de colas que se han migrado desde versiones anteriores a IBM MQ 8.0, pero se ha establecido en *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* para gestores de colas recién creados. Esta definición de **AUTHINFO** predeterminada tiene **CHCKCLNT** establecido en REQDADM de forma predeterminada.

Por lo tanto, los clientes existentes que utilizan un ID de usuario privilegiado para conectarse deben proporcionar credenciales válidas.

Aviso: Las credenciales de una estructura MQCSP para una aplicación cliente se envían a veces a través de la red en texto sin formato. Para asegurarse de que las credenciales de cliente están protegidas, consulte "Protección por contraseña MQCSP" en la página 32.

Granularidad de la configuración

Los atributos **CHCKLOCL** y **CHCKCLNT** del objeto AUTHINFO establecen los requisitos de autenticación para todas las conexiones con el gestor de colas. Además de estos atributos, el atributo **CHCKCLNT** en las reglas de autenticación de canal (CHLAUTH) permite establecer requisitos de autenticación más estrictos para conexiones de cliente específicas que coinciden con la regla CHLAUTH.

Puede establecer el valor global de **CHCKCLNT** en OPTIONAL, por ejemplo, en el objeto AUTHINFO y, a continuación, actualizarlo para que sea más estricto para determinados canales estableciendo **CHCKCLNT** en REQUIRED o REQDADM en la regla CHLAUTH. De forma predeterminada, las reglas CHLAUTH se definen con **CHCKCLNT(ASQMGR)**, por lo que no es necesario utilizar esta granularidad. Por ejemplo, estos mandatos MQSC definen una regla CHLAUTH que altera temporalmente el atributo **CHCKCLNT** del objeto AUTHINFO y una regla CHLAUTH que no:

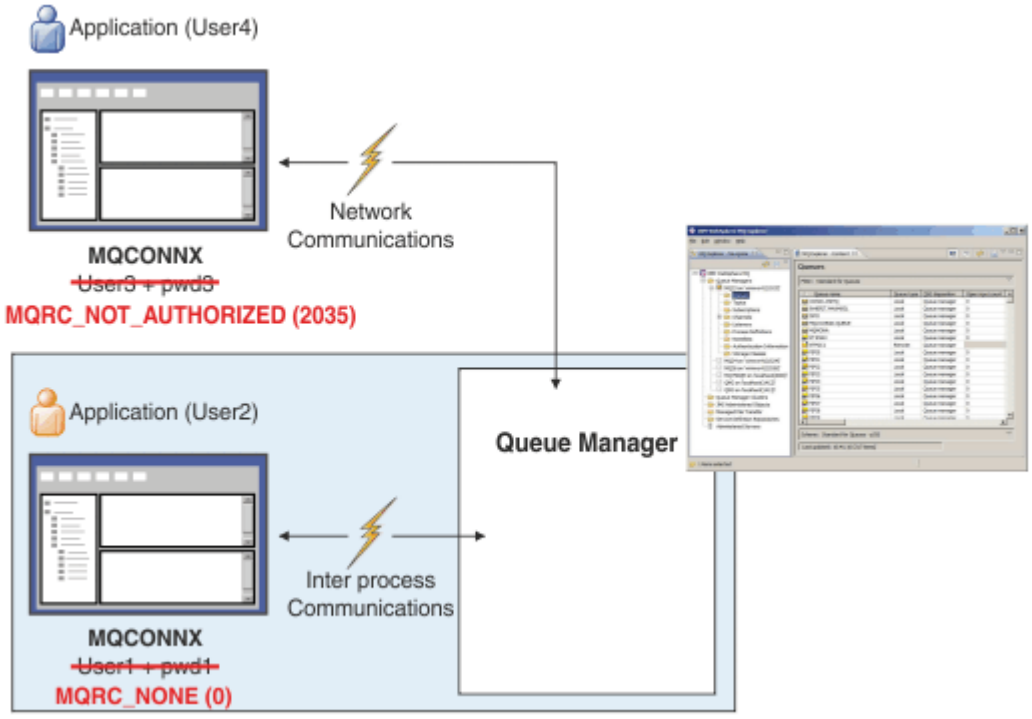
```
DEFINE AUTHINFO(USE.PW) AUTHTYPE(xxxxxx) +
CHCKCLNT(OPTIONAL)

SET CHLAUTH('*') TYPE(ADDRESSMAP) +
ADDRESS('*') USERSRC(CHANNEL) +
CHCKCLNT(REQUIRED)
```

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

Para obtener más información sobre las reglas CHLAUTH, consulte “Registros de autenticación de canal” en la página 53.

Notificación de errores



Se registra un error en las situaciones siguientes:

- Una aplicación no proporciona credenciales de autenticación cuando son necesarias.
- Una aplicación proporciona credenciales de autenticación no válidas. Esta situación se trata como un error incluso si la configuración indica que es opcional que las aplicaciones suministren credenciales.

Nota: Cuando **CHCKLOCL** o **CHCKCLNT** se establece en NONE, no se detectan las credenciales no válidas proporcionadas por las aplicaciones.

Las autenticaciones con errores se conservan durante el número de segundos especificado en el atributo **FAILDLAY** antes de que el error se devuelva a la aplicación. Este retardo proporciona cierta protección frente a una aplicación que intenta conectarse repetidamente.

El error se registra de varias maneras:

Aplicación
Se devuelve un código de razón MQRC_NOT_AUTHORIZED (2035) a la aplicación.

Administrador
Un administrador de IBM MQ ve el suceso notificado en el registro de errores. El mensaje de error muestra que la conexión se ha rechazado porque las credenciales no son válidas, en lugar de porque, por ejemplo, el usuario no tiene autorización de conexión.

Herramienta de supervisión
Una herramienta de supervisión también se puede notificar de la anomalía, si activa sucesos de autorización, mediante un mensaje de suceso en la cola SYSTEM.ADMIN.QMGR.EVENT . Para activar los sucesos de autorización, emita el siguiente mandato MQSC:

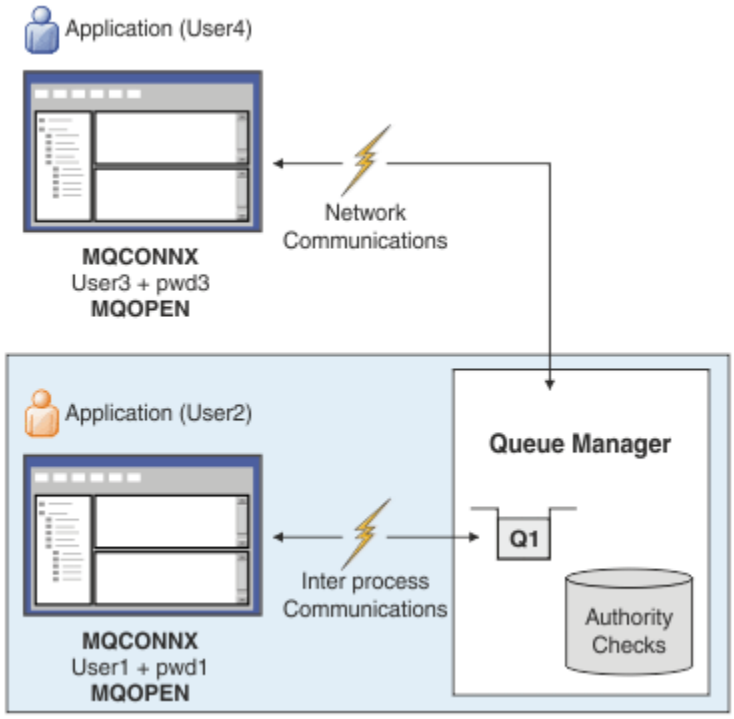
```
ALTER QMGR AUTHOREV(ENABLED)
```

Este suceso "No autorizado" es un suceso de conexión de tipo 1 y proporciona los mismos campos que otros sucesos de tipo 1, con un campo adicional, el ID de usuario MQCSP que se ha proporcionado. Si la aplicación ha proporcionado una contraseña, no se incluye en el mensaje de suceso. Esto significa que hay dos ID de usuario en el mensaje de suceso:

- El ID de usuario con el que se ejecuta la aplicación.
- El ID de usuario en las credenciales que ha presentado la aplicación.

Para obtener más información sobre este mensaje de suceso, consulte No autorizado (tipo 1).

Adopción de usuarios para autorización



Puede configurar el gestor de colas para que adopte las credenciales presentadas por la aplicación como contexto para la conexión. La adopción de las credenciales significa que el ID de usuario proporcionado en las credenciales de autenticación se utiliza para las comprobaciones de autorización, se muestra en las pantallas administrativas y aparece en los mensajes. El atributo **ADOPTCTX** en el objeto AUTHINFO controla si las credenciales se adoptan como contexto para la aplicación. Por ejemplo, los siguientes mandatos MQSC definen un objeto AUTHINFO denominado USE.PWD que se utiliza para la autenticación de conexión y establecen el atributo **ADOPTCTX** en YES:

```
DEFINE AUTHINFO(USE.PWD) +
AUTHTYPE(xxxxxx) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED) +
ADOPTCTX(YES)

ALTER QMGR CONNAUTH(USE.PWD)
```

Se pueden especificar los valores siguientes para el atributo **ADOPTCTX** :

ADOPTCTX(YES)
Las credenciales proporcionadas por la aplicación se adoptan como el contexto de aplicación durante la duración de la conexión. Todas las comprobaciones de autorización para una aplicación se realizan con el ID de usuario en las credenciales que se han autenticado.

Atención: Al utilizar **ADOPTCTX(YES)** y los ID de usuario del sistema operativo local, debe asegurarse de que el ID de usuario que se está adoptando cumple los requisitos para los ID de usuario en IBM MQ. Para obtener más información, consulte “ID de usuario” en la página 93.

ADOPTCTX(NO)
Las credenciales proporcionadas por una aplicación sólo se utilizan para la autenticación en el momento de la conexión. El ID de usuario con el que se ejecuta la aplicación se sigue utilizando para futuras comprobaciones de autorización. Es posible que encuentre esta opción útil al migrar, o si tiene previsto utilizar otros mecanismos, como registros de autenticación de canal, para asignar el identificador de usuario del agente de canal de mensajes (MCAUSER).

Interacción con autenticación de canal

Las reglas de autenticación de canal se pueden utilizar para cambiar el ID de usuario que se utiliza como contexto para una conexión de aplicación, basándose en el ID de usuario recibido del cliente. Para ver un ejemplo de cómo utilizar una regla de autenticación de canal para cambiar el ID de usuario asociado a una conexión, consulte "Correlación de un ID de usuario cliente con un ID de usuario MCAUSER" en la página 394.

El orden en que se procesan las reglas de autenticación de conexión y de canal es un factor importante a la hora de determinar el contexto de seguridad de las conexiones de aplicación de cliente de IBM MQ. El parámetro **ChlauthEarlyAdopt** de la stanza **channels** del archivo `qm.ini` controla el orden en el que el gestor de colas adopta el contexto de las credenciales proporcionadas por la aplicación y aplica las reglas de autenticación de canal. Para obtener más información sobre **ChlauthEarlyAdopt**, consulte Atributos de la stanza de canales.

Atención: Cuando se utiliza el parámetro **ADOPTCTX(YES)** en el objeto de información de autenticación, el contexto que se adopta a partir de las credenciales proporcionadas por la aplicación sólo se puede cambiar mediante reglas de autenticación de canal si el parámetro **ChlauthEarlyAdopt** se establece en Y.

Para obtener más información sobre la interacción de la autenticación de conexión y la autenticación de canal, y el orden en el que tienen lugar las comprobaciones cuando una aplicación cliente se conecta a un gestor de colas, consulte "Interacción de CHLAUTH y CONNAUTH" en la página 60.

Conceptos relacionados

"Autenticación de conexión" en la página 74
La autenticación de conexión permite a las aplicaciones proporcionar credenciales de autenticación cuando se conectan a un gestor de colas. El gestor de colas valida las credenciales. El ID de usuario proporcionado en las credenciales también se puede adoptar para su uso en comprobaciones de autorización para los recursos a los que accede la aplicación.

"Autenticación de conexión: Cambios en la aplicación" en la página 80

"Autenticación de conexión: Depósitos de usuario" en la página 81
Para cada uno de los gestores de colas, puede seleccionar diferentes tipos de objetos de información de autenticación para autenticar los ID de usuario y las contraseñas.

Autenticación de conexión: Cambios en la aplicación

Una aplicación que utiliza la interfaz de cola de mensajes (MQI) puede proporcionar un ID de usuario y una contraseña en la estructura de parámetros de seguridad de conexión (MQCSP) cuando se llama a MQCONNX. En otras interfaces de programación de aplicaciones, la estructura MQCSP normalmente se construye en nombre de la aplicación mediante las bibliotecas IBM MQ .

V 9.4.0 A partir de IBM MQ 9.3.4, las aplicaciones cliente que se conectan a un gestor de colas que se ejecuta en sistemas AIX o Linux también pueden enviar una señal de autenticación en la estructura MQCSP como un medio alternativo de identificación.

El ID de usuario y la contraseña, o la señal de autenticación, se pasan para comprobar el gestor de autorizaciones sobre objetos (OAM) proporcionado con el gestor de colas, o el componente de servicio de autorización proporcionado con el gestor de colas en sistemas z/OS . No es necesario escribir una interfaz personalizada.

Si la aplicación se ejecuta como un cliente, el ID de usuario y la contraseña, o la señal de autenticación, también se pasa a las salidas de seguridad del lado del cliente y del lado del servidor para su proceso.

También se pueden utilizar para establecer el atributo de identificador de usuario de agente de canal de mensajes (MCAUSER) de una instancia de canal.

Aviso: Las credenciales de una estructura MQCSP para una aplicación cliente se envían a veces a través de la red en texto sin formato. Para asegurarse de que las credenciales de aplicación cliente están protegidas, consulte “Protección por contraseña MQCSP” en la página 32.

Al utilizar la serie XAOPEN para proporcionar un ID de usuario y una contraseña, puede evitar tener que cambiar el código de aplicación.

Nota:

A partir de IBM WebSphere MQ 6.0, la salida de seguridad permite establecer MQCSP. Por lo tanto, los clientes en este nivel o posterior no tienen que actualizarse.

No obstante, en versiones de IBM MQ anteriores a la IBM MQ 8.0, MQCSP no imponía ninguna restricción en el ID de usuario y la contraseña proporcionados por la aplicación. Al utilizar estos valores con características proporcionadas por IBM MQ hay límites que se aplican a la utilización de estas características, pero si sólo las está pasando a sus propias salidas, esos límites no se aplican.

Conceptos relacionados

“Autenticación de conexión” en la página 74
La autenticación de conexión permite a las aplicaciones proporcionar credenciales de autenticación cuando se conectan a un gestor de colas. El gestor de colas valida las credenciales. El ID de usuario proporcionado en las credenciales también se puede adoptar para su uso en comprobaciones de autorización para los recursos a los que accede la aplicación.

“Autenticación de conexión: Configuración” en la página 75
Un gestor de colas se puede configurar para autenticar las credenciales proporcionadas por una aplicación cuando se conecta.

“Autenticación de conexión: Depósitos de usuario” en la página 81
Para cada uno de los gestores de colas, puede seleccionar diferentes tipos de objetos de información de autenticación para autenticar los ID de usuario y las contraseñas.

Autenticación de conexión: Depósitos de usuario

Para cada uno de los gestores de colas, puede seleccionar diferentes tipos de objetos de información de autenticación para autenticar los ID de usuario y las contraseñas.

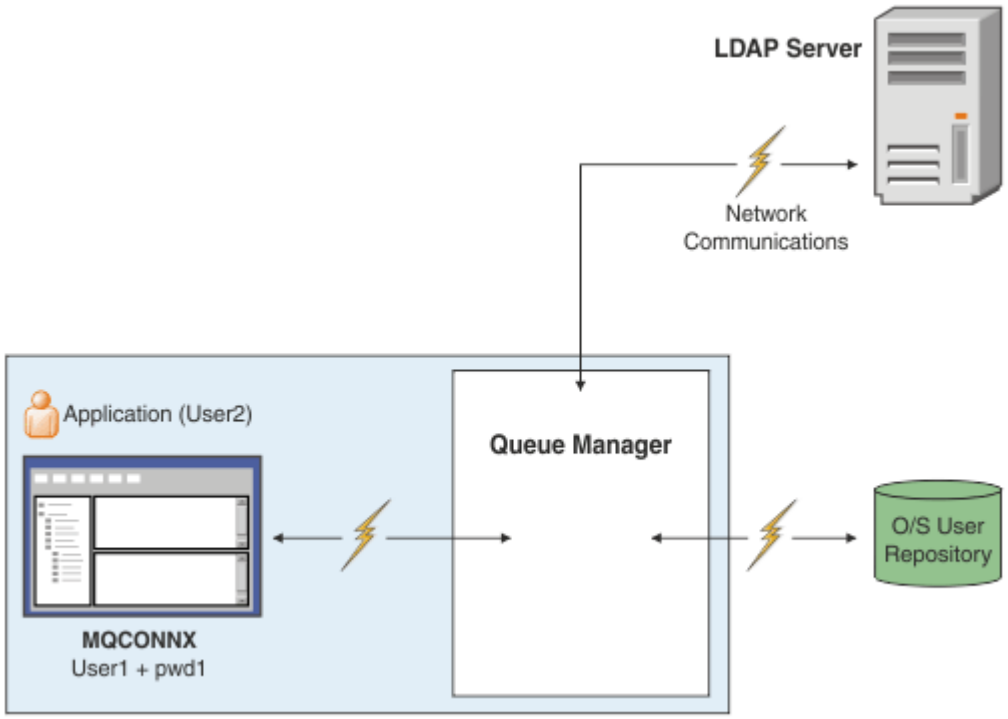


Figura 7. Tipos de objetos de información de autenticación

```
DEFINE AUTHINFO(USE.OS) AUTHTYPE(IDPWOS)
DEFINE AUTHINFO(USE.LDAP) +
AUTHTYPE(IDPWLDAP) +
CONNAME('ldap1(389),ldap2(389)') +
LDAPUSER('CN=QMGR1') +
LDAPPWD('passw0rd') SECCOMM(YES)
```

Existen dos tipos de objetos de información de autenticación, según se representa en el diagrama:

- `IDPWOS` se utiliza para indicar que el gestor de colas utiliza el sistema operativo local para autenticar el ID de usuario y la contraseña. Si opta por utilizar el sistema operativo local, necesita establecer los atributos comunes, tal como se describe en los temas anteriores.
- `IDPWLDAP` se utiliza para indicar que el gestor de colas utiliza un servidor LDAP para autenticar el ID de usuario y la contraseña. Si opta por utilizar un servidor LDAP, se proporciona más información en este tema.

Solo se puede seleccionar un objeto de información de autenticación para que lo utilice cada gestor de colas, especificando el nombre del objeto correspondiente en el atributo **CONNAUTH** del gestor de colas.

Utilización de un servidor LDAP para la autenticación.

Establezca el campo **CONNAME** en la dirección del servidor LDAP del gestor de colas. Puede proporcionar direcciones adicionales para el servidor LDAP en la lista separada por comas, lo que puede ayudarle si existen redundancias cuando el servidor LDAP no proporciona por su cuenta este recurso.

Establezca el ID de servidor y la contraseña LDAP en los campos **LDAPUSER** y **LDAPPWD**, de modo que el gestor de colas pueda acceder al servidor LDAP y buscar información acerca de los registros de usuario.

Conexión segura con un servidor LDAP

A diferencia de los canales, no existe ningún parámetro **SSLCIPH** que active el uso de TLS para la comunicación con el servidor LDAP. En este caso, IBM MQ actúa como cliente para el servidor LDAP, por lo que gran parte de la configuración se realiza en el servidor LDAP. Algunos parámetros existentes en IBM MQ se utilizan para configurar el modo en que funciona la conexión.

Establezca el campo **SECCOMM** para controlar si la conectividad con el servidor LDAP utiliza TLS.

Además de este atributo, los atributos del gestor de colas **SSLFIPS** y **SUITEB** restringen el conjunto de especificaciones de cifrado que se seleccionan. El certificado que se utiliza para identificar el gestor de colas en el servidor LDAP es el certificado del gestor de colas, ya sea `ibmwebspheremq` *`qmgr-name`* o el valor del atributo **CERTLABL** . Para obtener información detallada, consulte Etiquetas de certificados digitales.

Depósito de usuario LDAP

Si se utiliza un depósito de usuario LDAP, es necesario realizar más pasos de configuración en el gestor de colas que simplemente indicar el gestor de colas donde se encuentra el servidor AP.

Los ID de usuario definidos en un servidor LDAP tienen una estructura jerárquica que los identifica de forma exclusiva. Por lo tanto, una aplicación se puede conectar al gestor de colas y presentar su ID de usuario como el ID de usuario jerárquico totalmente calificado.

No obstante, para simplificar la información que debe proporcionar una aplicación, se puede configurar un gestor de colas que asuma que la primera parte de la jerarquía es común a todos los ID y que la añada automáticamente antes del ID abreviado que proporciona la aplicación. A continuación, el gestor de colas puede presentar un ID completo al servidor LDAP.

Establezca BASEDNU en el punto inicial en que la búsqueda LDAP busca el ID en la jerarquía de LDAP. Cuando haya establecido BASEDNU, debe asegurarse de que solo se devuelve un resultado al buscar el ID en la jerarquía de LDAP.

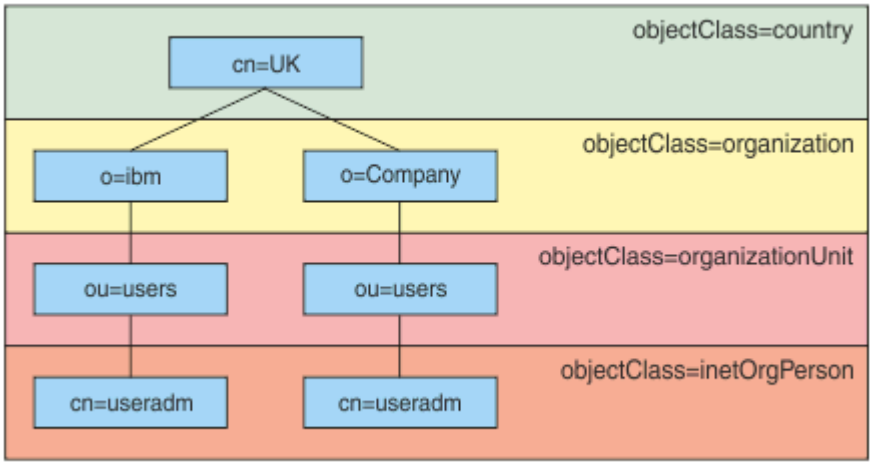


Figura 8. Un ejemplo de jerarquía de LDAP

Por ejemplo, en Figura 8 en la página 83, BASEDNU se puede establecer en "ou=users,o=ibm,c=UK" o en ",o=ibm,c=UK". No obstante, debido a que existe un nombre distinguido que contiene "cn=useradm" en la rama "o=ibm" y en la rama "o=Company", BASEDNU no se puede establecer en "c=UK". Por motivos de rendimiento y seguridad, utilice el punto más alto de la jerarquía de LDAP desde el que pueda hacer referencia a todos los ID de usuario que necesite. En este ejemplo, sería "ou=users,o=ibm,c=UK".

Su aplicación puede enviar al gestor de colas el ID de usuario sin proporcionar el nombre de atributo LDAP, por ejemplo, CN=. Si establece USRFIELD como de nombre de atributo LDAP, se añade este valor como prefijo del ID de usuario que procede de la aplicación. Esto puede resultar útil como ayuda para la migración cuando se mueven los ID de usuario del sistema operativo a los ID de usuario LDAP, ya que la aplicación puede presentar entonces la misma serie en ambos casos y evitará tener que cambiar la aplicación.

Por lo tanto, el ID de usuario completo presentado al servidor LDAP será similar al siguiente:

```
USRFIELD = ID_from_application BASEDNU
```

Conceptos relacionados

"Autenticación de conexión" en la página 74
La autenticación de conexión permite a las aplicaciones proporcionar credenciales de autenticación cuando se conectan a un gestor de colas. El gestor de colas valida las credenciales. El ID de usuario proporcionado en las credenciales también se puede adoptar para su uso en comprobaciones de autorización para los recursos a los que accede la aplicación.

"Autenticación de conexión: Configuración" en la página 75
Un gestor de colas se puede configurar para autenticar las credenciales proporcionadas por una aplicación cuando se conecta.

"Autenticación de conexión: Cambios en la aplicación" en la página 80

Salida de seguridad del lado del cliente para insertar ID de usuario y contraseña (mqccred)

Si tiene aplicaciones cliente que son necesarias para enviar un ID de usuario o una contraseña pero aún no puede cambiar el origen, existe una salida de seguridad que se entrega con IBM MQ 8.0 llamada **mqccred** que puede utilizar. **mqccred** proporciona un ID de usuario y una contraseña en nombre de la aplicación cliente, en un archivo `.ini`. Este ID de usuario y contraseña se envían al gestor de colas que, si se configura para ello, los autenticará.

Visión general

mqccred es una salida de seguridad que se ejecuta en la misma máquina que la aplicación cliente. Permite suministrar información de ID de usuario y contraseña en nombre de la aplicación cliente, donde dicha información no la está proporcionando la propia aplicación. La información de ID de usuario y contraseña se proporciona en una estructura conocida como Parámetros de seguridad de conexión (MQCSP) y la autenticará el gestor de colas si se ha configurado la autenticación de conexión.

La información de ID de usuario y contraseña se recupera de un archivo `.ini` en la máquina cliente. Las contraseñas del archivo se protegen mediante enmascaramiento mediante el mandato **runmqccred** y también asegurándose de que los permisos de archivo del archivo `.ini` se han establecido de modo que solo el ID de usuario que ejecuta la aplicación cliente (y por lo tanto la salida) pueda leerlo.

Ubicación

mqccred está instalado:

Plataformas Windows
En el directorio *installation_directory*\Tools\c\Samples\mqccred\

Plataformas AIX and Linux
En el directorio *installation_directory*/samp/mqccred

Notas: La salida:

1. Funciona exclusivamente como una salida de canal de seguridad y debe ser la única salida de este tipo definida en un canal.
2. Normalmente se nombra a través de la tabla de definiciones de canal de cliente (CCDT), pero un cliente Java puede tener la salida especificada directamente en los objetos JNDI, o se puede configurar la salida para las aplicaciones que construyen manualmente la estructura MQCD.
3. Debe copiar los programas **mqccred** y **mqccred_r** en el directorio `var/mqm/exits`.

   Por ejemplo, en un sistema AIX o Linux de 64 bits, emita el mandato:

   ```
   cp installation_directory/samp/mqccred/lib64/* /var/mqm/exits
   ```

   Para obtener más información, consulte Un ejemplo paso a paso de cómo probar mqccred.
4. Es capaz de ejecutarse en versiones anteriores de IBM MQ, tan lejos como IBM WebSphere MQ 7.0.1.

Configuración de ID de usuario y contraseñas

El archivo `.ini` contiene stanzas para cada gestor de colas con un valor global para gestores de colas no especificados. Cada stanza contiene el nombre del gestor de colas, un ID de usuario y una contraseña de texto sin formato o enmascarada.

Debe editar el archivo `.ini` manualmente, utilizando el editor que desee, y añada el atributo de contraseña de texto sin formato a las stanzas. Ejecute el programa **runmqccred** proporcionado, que toma el archivo `.ini` y sustituye el atributo **Password** por el atributo **OPW** , un formato ofuscado de la contraseña.

Consulte runmqccred para obtener una descripción del mandato y sus parámetros.

El archivo `mqccred.ini` contiene la información de ID de usuario y contraseña.

Se proporciona un archivo `.ini` de plantilla en el mismo directorio que la salida para proporcionar un punto de partida para la empresa.

De forma predeterminada, este archivo se buscará en `$HOME/.mqs/mqccred.ini`. Si desea ubicarlo en otro lugar, puede utilizar la variable de entorno *MQCCRED* de modo que apunte al mismo:

```
MQCCRED=C:\mydir\mqccred.ini
```

Si utiliza *MQCCRED*, la variable debe incluir el nombre completo del archivo de configuración, incluido todos los tipos de archivo `.ini`. Puesto que este archivo contiene contraseñas (incluso si están enmascaradas), se espera que se proteja el archivo utilizando privilegios de sistema operativo para asegurarse de que personas no autorizadas no puedan leerlo. Si no tiene el permiso de archivo correcto, la salida no se ejecutará satisfactoriamente.

Si la aplicación ya ha proporcionado una estructura MQCSP, la salida normalmente lo respeta y no insertará ninguna información del archivo `.ini`. No obstante, se puede alterar temporalmente utilizando el atributo **Force** en la stanza.

Si se establece **Force** en el valor *TRUE* se eliminará el ID de usuario y la contraseña proporcionados por la aplicación y se sustituirán por la versión del archivo ini.

También puede establecer el atributo **Force** en la sección global del archivo para establecer el valor predeterminado de dicho archivo.

El valor predeterminado para **Force** es *FALSE*.

Puede proporcionar un ID de usuario y contraseña para todos los gestores de colas, o para cada gestor de colas individual. A continuación se muestra un ejemplo de un archivo `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. Las definiciones de gestor de colas individuales tienen prioridad sobre el valor global.
2. Los atributos son sensibles a las mayúsculas y minúsculas.

Restricciones

Cuando esta salida está en uso, el ID de usuario local de la persona que ejecuta la aplicación no fluye del cliente al servidor. La única información de identidad disponible es desde el contenido de archivos ini.

Por lo tanto, debe configurar el gestor de colas para que utilice **ADOPTCTX(YES)**, o correlacione la solicitud de conexión de entrada con un ID de usuario adecuado a través de uno de los mecanismos disponibles, por ejemplo "Registros de autenticación de canal" en la página 53.

Importante: Si añade nuevas contraseñas, o actualiza contraseñas antiguas, el mandato **runmqccred** solo procesa todas las contraseñas de texto sin formato, dejando las enmascaradas sin modificar.

Depuración

La salida graba en el rastreo IBM MQ estándar cuando esta habilitado.

Para ayudarle a depurar los problemas de configuración, la salida también puede grabar directamente en la salida estándar.

Normalmente no se requiere ninguna configuración de dato de salida de seguridad de canal (**SCYDATA**) para el canal. Sin embargo, puede especificar:

ERROR
Imprime solo información sobre condiciones de error, como por ejemplo que no se puede encontrar el archivo de configuración.

DEBUG
Visualiza estas condiciones de error y algunas sentencias de rastreo adicionales.

NOCHECKS
Ignora las limitaciones sobre permisos de archivos y la limitación adicional que el archivo `.ini` no debe contener ninguna contraseña no protegida.

Puede poner uno o más de estos elementos en el campo **SCYDATA**, separados por comas, en cualquier orden. Por ejemplo, `SCYDATA=(NOCHECKS,DEBUG)`.

Tenga en cuenta que los elementos son sensibles a las mayúsculas y minúsculas, y deben especificarse en mayúsculas.

Utilización de `mqccred`

Una vez que ha configurado el archivo, puede invocar la salida de canal actualizando la definición de canal de conexión de cliente de modo que incluya el atributo `SCYEXIT('mqccred(ChlExit)')`:

```
DEFINE CHANNEL(channelname) CHLTYPE(clntconn) +
CONNAME(remote machine) +
QMNAME(remote qmgr) +
SCYEXIT('mqccred(ChlExit)') +
REPLACE
```

Referencia relacionada
SCYDATA
SCYEXIT
runmqccred

Autenticación de conexión con el cliente Java

La autenticación de conexión es una característica de IBM MQ que le permite configurar gestores de colas para que el gestor de colas pueda autenticar aplicaciones utilizando un ID de usuario y una contraseña proporcionados. Cuando la aplicación es una aplicación Java que utiliza el transporte de cliente, la autenticación de conexión se puede ejecutar en modalidad de compatibilidad o en modalidad de autenticación MQCSP.

La aplicación especifica el ID de usuario y la contraseña que se van a autenticar utilizando uno de los métodos siguientes:

- En una aplicación IBM MQ classes for Java , en la clase `MQEnvironment` , o las propiedades Hashtable que se pasan al constructor `com.ibm.mq.MQQueueManager` .
- En una aplicación IBM MQ classes for JMS , como argumentos para el método `createConnection(String username, String Password)` o `createContext(String username, String password)` .

Modalidad de autenticación MQCSP

En esta modalidad, el ID de usuario del lado del cliente bajo el que se ejecuta la aplicación se envía al gestor de colas, así como el ID de usuario y la contraseña que se van a autenticar. IBM MQ classes for Java y IBM MQ classes for JMS envían el ID de usuario y la contraseña que se van a autenticar al gestor de colas en una estructura MQCSP .

El ID de usuario y la contraseña están disponibles para una salida de seguridad de conexión de servidor dentro de la estructura MQCSP. La dirección de estructura MQCSP se puede encontrar en el campo **SecurityParms** de la estructura MQCXP para el canal.

La modalidad de autenticación MQCSP tiene las ventajas siguientes:

- La longitud máxima del ID de usuario que se va a autenticar es de 1024 caracteres.
- La longitud máxima de la contraseña para la autenticación es de 256 caracteres.
- Las comprobaciones de autorización para acceder a los recursos de IBM MQ se pueden realizar utilizando el ID de usuario del lado del cliente con el que se ejecuta la aplicación, cuando el objeto de información de autenticación que se utiliza para controlar la autenticación de conexión en el gestor de colas se configura con ADOPTCTX (NO).

Modalidad de compatibilidad

Antes de IBM MQ 8.0, el cliente Java podía enviar un ID de usuario y una contraseña por el canal de conexión de cliente al canal de conexión del servidor, y hacer que se suministren a una salida de seguridad en los campos **RemoteUserIdentifier** y **RemotePassword** de la estructura MQCD. En modalidad de compatibilidad, este comportamiento se retiene.

Puede utilizar esta modalidad en combinación con la autenticación de conexión y realizar la migración fuera de las salidas de seguridad que se utilizaron para realizar el mismo trabajo.

Esta modalidad tiene las restricciones siguientes:

- La longitud del ID de usuario y la contraseña debe ser de 12 caracteres o menos. Los ID de usuario de más de 12 caracteres se truncan a 12 caracteres. Esto puede hacer que la conexión falle con el código de razón MQRC_NOT_AUTHORIZED.
- El ID de usuario del lado del cliente bajo el que se ejecuta la aplicación no se envía al gestor de colas. Debe establecer ADOPTCTX (YES) en el objeto de información de autenticación que se utiliza para controlar la autenticación de conexión en el gestor de colas, o utilizar otro método, como una regla de autenticación de canal basada en un certificado TLS, para establecer el ID de usuario de MCA de canal que se comprueba para la autorización para utilizar recursos de IBM MQ .

Modalidad de autenticación predeterminada

La modalidad de autenticación predeterminada que utiliza una aplicación cliente IBM MQ classes for Java o IBM MQ classes for JMS varía en función de si la aplicación especifica un ID de usuario y una contraseña.

- Si se especifica un ID de usuario y una contraseña, se utiliza la autenticación MQCSP de forma predeterminada.
- Si se especifica un ID de usuario, pero no se especifica ninguna contraseña, se utiliza la modalidad de compatibilidad de forma predeterminada.
- Si no se especifica ningún ID de usuario, siempre se utiliza la modalidad de compatibilidad.

En los casos en los que se especifica un ID de usuario, la aplicación puede elegir una modalidad de autenticación específica para cada conexión individual, o se puede establecer globalmente antes de que se inicie la aplicación, tal como se describe en "Elección de la modalidad de autenticación" en la página 88.

Nota: Las aplicaciones que utilizan IBM MQ classes for JMS pueden verse afectadas por el cambio a la modalidad de autenticación predeterminada en IBM MQ 9.3.0. Después de actualizar IBM MQ classes for JMS a IBM MQ 9.3.0, las aplicaciones que anteriormente utilizaban la modalidad de compatibilidad de forma predeterminada utilizarán la autenticación MQCSP en su lugar. Esto puede hacer que las aplicaciones que anteriormente se conectaban correctamente a un gestor de colas no se conecten con un `JMSException` que contiene el código de razón 2035 (MQRC_NOT_AUTHORIZED). Si esto ocurre, utilice uno de los métodos descritos en "Elección de la modalidad de autenticación" en la página 88 para especificar que la aplicación utiliza la modalidad de compatibilidad.

Las aplicaciones Java que se conectan al gestor de colas utilizando enlaces locales siempre utilizan la modalidad de autenticación MQCSP.

Elección de la modalidad de autenticación

La modalidad de autenticación que utilizan las aplicaciones cliente de Java que especifican un ID de usuario al conectarse al gestor de colas se puede especificar utilizando uno de los métodos siguientes. Estos métodos se listan en orden decreciente de prioridad. Si no se especifica la modalidad de autenticación utilizando alguno de estos métodos, se utiliza la modalidad de autenticación predeterminada.

Nota: El uso de estos métodos para seleccionar el modo de autenticación se ha aclarado en IBM MQ 9.3.0. En algunos casos, el modo de autenticación utilizado por una aplicación cliente Java puede cambiar cuando IBM MQ classes for Java o IBM MQ classes for JMS se actualizan a IBM MQ 9.3.0. Esto puede hacer que las aplicaciones que anteriormente se conectaban correctamente a un gestor de colas no se conecten con un `JMSException` que contiene el código de razón 2035 (MQRC_NOT_AUTHORIZED). Si esto ocurre, utilice uno de los métodos siguientes para seleccionar el modo de autenticación necesario.

- Especifique la modalidad de autenticación para cada conexión individual estableciendo la propiedad adecuada en la aplicación antes de conectarse al gestor de colas.
  - Cuando utilice IBM MQ classes for Java, establezca la propiedad *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* en las propiedades Hashtable que se pasan al constructor `com.ibm.mq.MQQueueManager`.
  - Cuando utilice IBM MQ classes for JMS, establezca la propiedad *JmsConstants.USER_AUTHENTICATION_MQCSP* en la fábrica de conexiones adecuada antes de crear la conexión.

  Establezca el valor de estas propiedades en uno de los valores siguientes:

  true
  Utilice la modalidad de autenticación MQCSP al autenticar con un gestor de colas.

  falso
  Utilice la modalidad de compatibilidad al autenticar con un gestor de colas.
- Especifique la modalidad de autenticación para todas las conexiones de cliente realizadas por una aplicación estableciendo la propiedad del sistema *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java al iniciar la aplicación. Establezca el valor de la propiedad en uno de los valores siguientes:

  Y
  Utilice la modalidad de autenticación MQCSP al autenticar con un gestor de colas.

  N
  Utilice la modalidad de compatibilidad al autenticar con un gestor de colas.

  Por ejemplo, el mandato siguiente establece la propiedad para seleccionar la modalidad de compatibilidad e inicia una aplicación Java :

  ```
  java -Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=N application_name
  ```

- Especifique la modalidad de autenticación para todas las conexiones de cliente realizadas por aplicaciones iniciadas en el mismo entorno estableciendo la variable de entorno *com.ibm.mq.jmqi.useMQCSPauthentication* en el entorno donde se inicia la aplicación. Establezca el valor de la variable de entorno en uno de los valores siguientes:

  Y
  Utilice la modalidad de autenticación MQCSP al autenticar con un gestor de colas.

  N
  Utilice la modalidad de compatibilidad al autenticar con un gestor de colas.
- Especifique la modalidad de autenticación para todas las aplicaciones que utilizan un archivo de configuración de cliente IBM MQ MQI client específico especificando el atributo **useMQCSPauthentication** en la stanza JMQI del archivo de configuración de cliente. Establezca el valor del atributo en uno de los valores siguientes:

  SÍ
  Utilice la modalidad de autenticación MQCSP al autenticar con un gestor de colas.

  NO
  Utilice la modalidad de compatibilidad al autenticar con un gestor de colas.

  Para obtener más información sobre el atributo **useMQCSPauthentication** , consulte Stanza JMQI del archivo de configuración de cliente.

Selección del modo de autenticación en IBM MQ Explorer

IBM MQ Explorer es una aplicación Java, de forma que estas dos modalidades, modalidad de compatibilidad y modalidad de autenticación MQCSP, también le son aplicables.

La modalidad de autenticación MQCSP es el valor predeterminado.

En paneles donde se proporciona la identificación de usuario, hay un recuadro de selección para habilitar o inhabilitar el modo de compatibilidad:

- De forma predeterminada, este recuadro de selección no está seleccionado. Para utilizar el modo de compatibilidad, seleccione esta casilla.

Conceptos relacionados

“Autenticación de conexión” en la página 74
La autenticación de conexión permite a las aplicaciones proporcionar credenciales de autenticación cuando se conectan a un gestor de colas. El gestor de colas valida las credenciales. El ID de usuario proporcionado en las credenciales también se puede adoptar para su uso en comprobaciones de autorización para los recursos a los que accede la aplicación.

“Autenticación de conexión: Cambios en la aplicación” en la página 80

“Autenticación de conexión: Depósitos de usuario” en la página 81
Para cada uno de los gestores de colas, puede seleccionar diferentes tipos de objetos de información de autenticación para autenticar los ID de usuario y las contraseñas.

Seguridad de mensajes en IBM MQ

La seguridad de mensajes en la infraestructura de IBM MQ se proporciona por Advanced Message Security.

Advanced Message Security (AMS) amplía los servicios de seguridad de IBM MQ para proporcionar funciones de firma y cifrado de los datos a nivel de mensaje. Los servicios ampliados garantizan que los datos de los mensajes no se han modificado entre el momento en que se colocaron originalmente en una cola y cuando se recuperaron. Además, AMS verifica que el emisor de los datos de un mensaje está autorizado para colocar mensajes firmados en una cola de destino.

Conceptos relacionados

“Advanced Message Security” en la página 610

Advanced Message Security (AMS) es un componente de IBM MQ que proporciona un alto nivel de protección para los datos confidenciales que fluyen a través de la red IBM MQ, aunque no afecta a las aplicaciones finales.

Planificación de los requisitos de seguridad

En esta colección de temas se explica lo que debe tener en cuenta al planificar la seguridad en un entorno IBM MQ.

Puede utilizar IBM MQ para una amplia gama de aplicaciones de diferentes plataformas. Los requisitos de seguridad serán probablemente diferentes para cada aplicación. Para algunas, la seguridad será un tema importante.

IBM MQ proporciona una serie de servicios de seguridad a nivel de enlace, incluyendo soporte para TLS (seguridad de la capa de transporte).

Debe tener en cuenta determinados aspectos de seguridad al planificar la instalación de IBM MQ:

- Multi En Multiplatforms, si ignora estas cuestiones y no hace nada, no podrá utilizar IBM MQ.
- z/OS En z/OS, el efecto de ignorar estos aspectos es que los recursos de IBM MQ no están protegidos. Es decir, todos los usuarios pueden acceder y modificar todos los recursos de IBM MQ.

autorización para administrar IBM MQ

Los administradores de IBM MQ necesitan autorización para:

- Emitir mandatos para administrar IBM MQ
- Utilizar IBM MQ Explorer
- IBM i Utilizar los paneles y mandatos administrativos de IBM i.
- z/OS Utilizar las operaciones y los paneles de control en z/OS
- z/OS Utilizar el programa de utilidad de IBM MQ, CSQUTIL, en z/OS
- z/OS Acceder a los conjuntos de datos de gestor de colas en z/OS

Si desea ver más información, consulte:

- ALW “Autorización para administrar IBM MQ en AIX, Linux, and Windows” en la página 408
- IBM i “Autorización para administrar IBM MQ en IBM i” en la página 95
- z/OS “Authority to administer IBM MQ on z/OS” en la página 96

autorización para trabajar con objetos de IBM MQ

Las aplicaciones pueden acceder a los objetos de IBM MQ siguientes emitiendo llamadas MQI:

- Gestores de colas
- Colas
- todos los Procesos
- Listas de nombres
- Temas

Las aplicaciones también pueden utilizar mandatos de Formato de mandatos programables (PCF) para acceder a estos objetos IBM MQ y para acceder también a canales y a objetos de información de autenticación. Estos objetos pueden ser protegidos por IBM MQ para que los ID de usuario asociados a las aplicaciones necesiten autorización para acceder a ellos.

Para obtener más información, consulte “Autorización para que las aplicaciones utilicen IBM MQ” en la página 98.

Seguridad de canal

Los ID de usuario asociados a los agentes de canal de mensajes (MCA) necesitan autorización para acceder a diferentes recursos de IBM MQ. Por ejemplo, un MCA debe poder conectarse a un gestor de colas. Si se trata de un MCA emisor, debe poder abrir la cola de transmisión para el canal. Si se trata de un MCA emisor, debe poder abrir las colas de destino. El ID de usuario asociados con aplicaciones que necesitan administrar canales, iniciadores de canal y escuchas necesitan autorización para utilizar los mandatos PCF pertinentes. Sin embargo, la mayoría de las aplicaciones no necesitan este tipo de acceso.

Para obtener más información, consulte “Autorización de canal” en la página 120.

Consideraciones adicionales

Debe tener en cuenta los siguientes aspectos de seguridad solamente si utiliza determinadas extensiones de funciones o de producto base de IBM MQ:

- “Seguridad para clústeres de gestores de colas” en la página 133
- “Seguridad para Publicación/Suscripción de IBM MQ” en la página 134

Planificación de la identificación y autenticación

Decida qué ID de usuario va a utilizar, cómo y en qué niveles desea aplicar controles de autenticación.

Debe decidir cómo va a identificar a los usuarios de las aplicaciones de IBM MQ, teniendo en cuenta que distintos sistemas operativos dan soporte a ID de usuario de longitudes diferentes. Puede utilizar registros de autenticación de canal para correlacionar de un ID de usuario a otro, o para especificar un ID de usuario basándose en algún atributo de conexión. Los canales de IBM MQ que utilizan TLS utilizan los certificados digitales como mecanismo para la identificación y autenticación. Cada certificado digital tiene un nombre distinguido de asunto que se puede correlacionar con identidades específicas utilizando registros de autenticación de canal. Además, los certificados de CA del repositorio de claves determinan qué certificados digitales se pueden utilizar para autenticar en IBM MQ. Para obtener más información, consulte:

- “Correlacionar un gestor de colas remoto con un ID de usuario MCAUSER” en la página 394
- “Correlación de un ID de usuario cliente con un ID de usuario MCAUSER” en la página 394
- “Correlacionar un Nombre distinguido SSL o TLS con un ID de usuario MCAUSER” en la página 395
- “Correlacionar una dirección IP con un ID de usuario MCAUSER” en la página 397

Planificación de la autenticación para una aplicación cliente

Puede aplicar controles de autenticación en cuatro niveles: en el nivel de comunicaciones, en las salidas de seguridad, con registros de autenticación de canal y en términos de la identificación que se ha pasado a una salida de seguridad.

Hay cuatro niveles de seguridad a tener en cuenta. El diagrama muestra un IBM MQ MQI client que está conectado a un servidor. La seguridad se aplica en cuatro niveles, tal como se describe en el texto siguiente. MCA es un Agente de canal de mensajes.

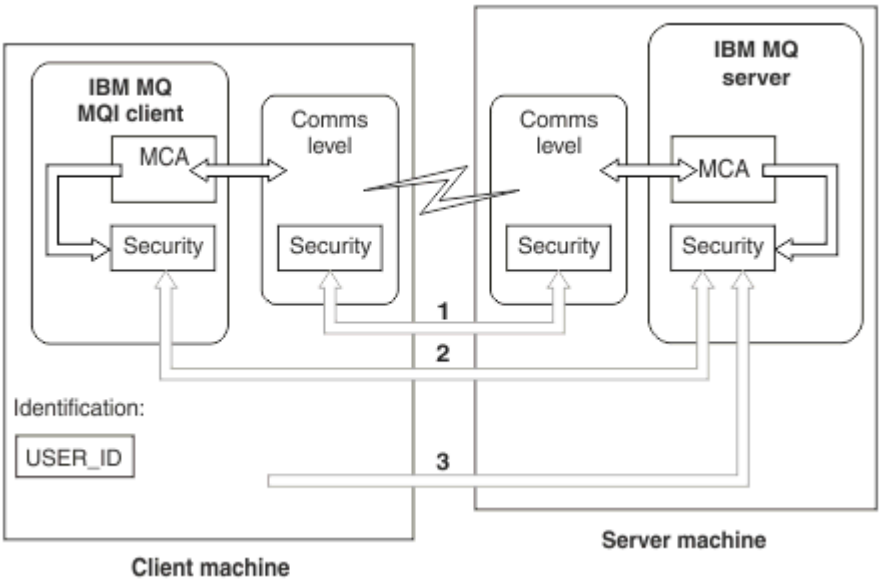


Figura 9. Seguridad en una conexión cliente/servidor

1. Nivel de comunicaciones

   Consulte la flecha 1. Para implementar la seguridad a nivel de comunicaciones, utilice TLS. Para obtener más información, consulte “Protocolos de seguridad de cifrado: TLS” en la página 18

2. Registros de autenticación de canal

   Consulte las flechas 2 y 3. La autenticación se puede controlar utilizando la dirección IP o los nombres distinguidos TLS en el nivel de seguridad. Un ID de usuario también se puede bloquear, o se puede correlacionar ID de usuario validado con un ID de usuario válido. En “Registros de autenticación de canal” en la página 53 se proporciona una descripción completa.

3. Autenticación de conexión

   Consulte la flecha 3. El cliente envía un ID de usuario y una contraseña, o una señal de autenticación. Para obtener más información, consulte “Autenticación de conexión: Configuración” en la página 75.

4. Salidas de seguridad de canal

   Consulte la flecha 2. Las salidas de seguridad de canal para la comunicación de cliente a servidor pueden funcionar de la misma forma que para la comunicación de servidor a servidor. Un par de salidas independientes del protocolo pueden escribirse para proporcionar la autenticación mutua tanto del cliente como del servidor. Se proporciona una descripción completa en Programas de salida de la seguridad de canal.

5. Identificación que se pasa a una salida de seguridad de canal.

   Consulte la flecha 3. En la comunicación de cliente a servidor, las salidas de seguridad de canal no tienen que funcionar como un par. La salida en el lado del cliente IBM MQ se puede omitir. En este caso, el ID de usuario se coloca en el descriptor del canal (MQCD) y la salida de seguridad del lado del servidor lo puede modificar, si es necesario.

   IBM MQ MQI clients también envía información adicional para ayudar a la identificación.

   - El ID de usuario que se pasa al servidor es el ID de usuario que está conectado actualmente al cliente.
   - El ID de seguridad del usuario conectado actualmente.

   Los valores del ID de usuario y, si está disponible, el ID de seguridad, pueden ser utilizados por la salida de seguridad del servidor para establecer la identidad del IBM MQ MQI client.

A partir de IBM MQ 8.0, puede enviar contraseñas incluidas en la estructura MQCSP.

Linux V 9.4.0 AIX A partir de IBM MQ 9.3.4, la conexión de IBM MQ MQI clients a gestores de colas IBM MQ que se ejecutan en sistemas AIX o Linux también puede enviar señales de autenticación en la estructura MQCSP.

Aviso: En algunos casos, la contraseña o señal de autenticación en una estructura MQCSP para una aplicación cliente se envía a través de la red en texto sin formato. Para asegurarse de que las contraseñas de aplicación cliente y las señales de autenticación están protegidas adecuadamente, consulte “Protección por contraseña MQCSP” en la página 32.

ID de usuario

Cuando crea ID de usuario para las aplicaciones de cliente, los ID de usuario no deben superar la longitud máxima permitida. No debe utilizar los ID de usuario reservados UNKNOWN y NOBODY. Si el servidor al que se conecta el cliente es un servidor IBM MQ for Windows , debe escapar el uso del signo de arroba, @. La longitud permitida de los ID de usuario depende de la plataforma que se utiliza para el servidor:

- Linux z/OS AIX En z/OS, AIX and Linux, la longitud máxima de un ID de usuario es de 12 caracteres.
- IBM i En IBM i, la longitud máxima de un ID de usuario es de 10 caracteres.
- Windows En Windows, si el servidor IBM MQ MQI client y el servidor IBM MQ están en Windows, y el servidor tiene acceso al dominio en el que está definido el ID de usuario de cliente, la longitud máxima de un ID de usuario es de 20 caracteres. No obstante, si el servidor de IBM MQ no es un servidor de Windows, el ID de usuario se trunca a 12 caracteres.
- Si utiliza la estructura MQCSP para pasar credenciales, la longitud máxima de un ID de usuario es de 1024 caracteres. El ID de usuario de la estructura MQCSP no se puede utilizar para eludir la longitud máxima de ID de usuario utilizada por IBM MQ para la autorización. Para obtener más información sobre la estructura MQCSP, consulte “Identificación y autenticación de usuarios utilizando la estructura MQCSP” en la página 328.

En sistemas AIX and Linux , el valor predeterminado es que los ID de usuario se utilizan para autenticarse y los grupos se utilizan para la autorización. Sin embargo, puede configurar estos sistemas para autorizarlos en los ID de usuario. Para obtener más información, consulte “Permisos basados en usuario de OAM en AIX and Linux” en la página 360. Los sistemas Windows pueden utilizar tanto los ID de usuario para la autenticación como para la autorización y los grupos para la autorización.

Si crea cuentas de servicio, sin prestar atención a los grupos y autoriza todos los ID de usuario de manera diferente, todos los usuarios pueden acceder a la información del resto de usuarios.

ID de usuario restringidos

Los ID de usuario UNKNOWN y el grupo NOBODY tienen significados especiales en IBM MQ. La creación de un ID de usuario en el sistema operativo denominado UNKNOWN o un grupo denominado NOBODY podría tener resultados no deseados.

Los ID de usuario cuando se conecta a un servidor de IBM MQ for Windows

Windows

Un servidor IBM MQ for Windows no da soporte a la conexión de un IBM MQ MQI client si el cliente se ejecuta con un ID de usuario que contiene el carácter @, por ejemplo, `abc@d`. El código de retorno para la llamada MQCONN en el cliente es MQRC_NOT_AUTHORIZED.

Sin embargo, puede especificar el ID de usuario utilizando dos caracteres @, por ejemplo, `abc@@d`. El uso del formato `id@domain` es la práctica preferida, para asegurarse de que el ID de usuario se resuelve en el dominio correcto de forma coherente; por lo tanto, `abc@@d@domain`.

Planificación de la autorización

Planifique los usuarios que tendrán autorización administrativa y planifique cómo autorizar a los usuarios de aplicaciones para que utilicen correctamente los objetos de IBM MQ incluidos los que se conectan desde un IBM MQ MQI client.

Para poder utilizar IBM MQ se debe otorgar acceso a personas o a aplicaciones. Qué acceso requieren dependerá de los roles que realicen y de las tareas que deban realizar. La autorización en IBM MQ puede subdividirse en dos categorías principales:

- Autorización para realizar operaciones administrativas
- Autorización para que las aplicaciones utilicen IBM MQ

Ambas clases de operación las controla el mismo componente y se puede otorgar autorización a una persona para que lleve a cabo las dos categorías de operación.

Los temas siguientes proporcionan más información sobre áreas de autorización específicas que debe tener en cuenta:

autorización para administrar IBM MQ

Los administradores de IBM MQ necesitan autorización para realizar diversas funciones. Esta autorización se obtiene de diferentes maneras en diferentes plataformas.

Los administradores de IBM MQ necesitan autorización para:

- Emitir mandatos para administrar IBM MQ.
- Windows Linux Utilizar IBM MQ Explorer.
- z/OS Utilizar las operaciones y los paneles de control en z/OS.
- z/OS Utilizar el programa de utilidad de IBM MQ, CSQUTIL, en z/OS.
- z/OS Acceder a los conjuntos de datos de gestor de colas en z/OS.

Para obtener más información, consulte el tema correspondiente a su sistema operativo.

ALW *Autorización para administrar IBM MQ en sistemas AIX, Linux, and Windows*

Un administrador de IBM MQ es un miembro del grupo mqm. Este grupo tiene acceso a todos los recursos de IBM MQ y puede emitir mandatos de control IBM MQ. Un administrador puede otorgar autorizaciones específicas a otros usuarios.

Para ser administrador de IBM MQ en sistemas AIX, Linux, and Windows , un usuario debe ser miembro del *grupo mqm*. Este grupo se crea automáticamente cuando se instala IBM MQ. Para permitir que los usuarios emitan mandatos de control, debe añadirlos al grupo mqm. Esto incluye el usuario root AIX and Linux.

A los usuarios que no son miembros del grupo mqm se les pueden otorgar privilegios administrativos, pero no pueden emitir mandatos de control de IBM MQ, y tienen autorización para ejecutar solamente los mandatos para los que se les ha otorgado acceso.

Además, en los sistemas Windows , las cuentas SYSTEM y Administrator tienen acceso completo a los recursos de IBM MQ .

Todos los miembros del grupo mqm tienen acceso a todos los recursos de IBM MQ del sistema, incluida la posibilidad de administrar cualquier gestor de colas que se ejecute en el sistema. Este acceso solamente se puede revocar si se suprime un usuario del grupo mqm. En los sistemas Windows, los miembros del grupo de administradores también tienen acceso a todos los recursos de IBM MQ.

Los administradores pueden utilizar el mandato **runmqsc** para emitir mandatos de script de IBM MQ (MQSC). Cuando se utiliza **runmqsc** en modalidad indirecta para enviar mandatos MQSC a un gestor de colas remoto, todo mandato MQSC se encapsula en un mandato PCF de escape. Los administradores

deben tener las autorizaciones necesarias para que el gestor de colas remoto procese los mandatos MQSC.

IBM MQ Explorer emite mandatos PCF para realizar tareas de administración. Los administradores no necesitan autorizaciones adicionales para utilizar IBM MQ Explorer para administrar un gestor de colas en el sistema local. Cuando IBM MQ Explorer se utiliza para administrar un gestor de colas en otro sistema, los administradores deben tener las autorizaciones necesarias para que el gestor de colas remoto procese los mandatos PCF.

Para obtener más información sobre las comprobaciones de autorización que se llevan a cabo cuando se procesan mandatos PCF y MQSC, consulte los temas siguientes :

- Para los mandatos que se ejecutan en gestores de colas, colas, canales, procesos, listas de nombres y objetos de información de autenticación, consulte "Autorización para que las aplicaciones utilicen IBM MQ" en la página 98.
- Para los mandatos que se ejecutan en canales, iniciadores de canal, escuchas y clústeres, consulte Seguridad de canal.
- z/OS Para los mandatos MQSC que procesa el servidor de mandatos en IBM MQ for z/OS, consulte "Command security and command resource security on z/OS" en la página 96.

Para obtener más información sobre la autorización que necesita para administrar sistemas IBM MQ for AIX, Linux, and Windows , consulte la información relacionada.

IBM i *Autorización para administrar IBM MQ en IBM i*

Para ser un administrador de IBM MQ en IBM i, debe ser miembro del *grupo QMQMADM*. Este grupo tiene propiedades similares a las del grupo mqm en sistemas AIX, Linux, and Windows . En particular, el grupo QMQMADM se crea al instalar IBM MQ for IBM i y los miembros del grupo QMQMADM tienen acceso a todos los recursos de IBM MQ en el sistema. También tiene acceso a todos los recursos IBM MQ si tiene autorización *ALLOBJ.

Los administradores pueden utilizar mandatos CL para administrar IBM MQ.Uno de estos mandatos es GRTMQMAUT, que se utiliza para conceder autorizaciones a otros usuarios. Otro mandato, STRMQMMQSC, permite que un administrador emita mandatos MQSC a un gestor de colas local.

Hay dos grupos de mandatos de CL proporcionados por IBM MQ for IBM i:

Grupo 1
Para emitir un mandato en esta categoría, un usuario debe ser miembro del grupo QMQMADM o tener autorización *ALLOBJ. Por ejemplo, GRTMQMAUT y STRMQMMQSC pertenecen a este categoría.

Grupo 2
Para emitir un mandato en esta categoría, un usuario no necesita ser miembro del grupo QMQMADM ni tener autorización *ALLOBJ. En su lugar, se necesitan dos niveles de autorización:

- El usuario necesita autorización de IBM i para utilizar el mandato. Esta autorización se concede mediante el mandato GRTOBJAUT.
- El usuario necesita autorización de IBM MQ para acceder a cualquier objeto IBM MQ asociado con el mandato. Esta autorización se concede mediante el mandato GRTMQMAUT.

Los ejemplos siguientes muestran mandatos de este grupo:

- CRTMQMQ, Crear cola MQM
- CHGMQMPRC, Cambiar proceso MQM
- DLTMQMNL, Suprimir lista de nombres MQM
- DSPMQMAUTI, Visualizar información de autenticación MQM
- CRTMQMCHL, Crear canal MQM

Para obtener más información sobre este grupo de mandatos, consulte el apartado "Autorización para que las aplicaciones utilicen IBM MQ" en la página 98.

Para obtener una lista completa de los mandatos de los grupos 1 y 2, consulte “Autorizaciones de acceso para los objetos de IBM MQ en IBM i” en la página 166

Para obtener más información sobre la autorización que necesita para administrar IBM MQ en IBM i, consulte Administración de 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.

Autorización para que las aplicaciones utilicen IBM MQ

Cuando las aplicaciones acceden a objetos, los ID de usuario asociados a las aplicaciones necesitan la autorización adecuada.

Las aplicaciones pueden acceder a los objetos de IBM MQ siguientes emitiendo llamadas MQI:

- Gestores de colas
- Colas
- todos los Procesos
- Listas de nombres
- Temas

Las aplicaciones también pueden utilizar mandatos PCF para administrar objetos de IBM MQ. Cuando se procesa el mandato PCF, utiliza el contexto de autorización del ID de usuario que ha transferido el mensaje PCF.

Las aplicaciones, en este contexto, incluyen las escritas por usuarios y proveedores, y las proporcionadas con IBM MQ for z/OS.

z/OS Las aplicaciones que se proporcionan con IBM MQ for z/OS son:

- Los paneles de operaciones y los paneles de control
- El programa de utilidad de IBM MQ, CSQUTIL
- El programa de utilidad del manejador de colas de mensajes no entregados, CSQUDLQH

Las aplicaciones que utilizan IBM MQ classes for Java, IBM MQ classes for JMS, IBM MQ classes for .NET o Message Service Clients for C/C++ y .NET utilizan indirectamente la interfaz MQI.

Los MCA también emiten llamadas MQI y los ID de usuario asociados a los MCA necesitan autorización para acceder a estos objetos de IBM MQ. Para obtener más información acerca de estos ID de usuario y las autorizaciones que necesitan, consulte “Autorización de canal” en la página 120.

z/OS En z/OS, las aplicaciones también utilizan mandatos MQSC para acceder a estos objetos de IBM MQ pero la seguridad de mandatos y la seguridad de recursos de mandatos proporcionan comprobaciones de autorización en estas circunstancias. z/OS Para obtener más información, consulte “Command security and command resource security on z/OS” en la página 96 and “MQSC commands and the system command input queue on z/OS” en la página 97.

IBM i En IBM i, un usuario que emite un mandato de CL del Grupo 2 podría necesitar autorización para acceder a un objeto de IBM MQ asociado al mandato. Para obtener más información, consulte “Cuándo se efectúan las comprobaciones de autorización” en la página 98.

Cuándo se efectúan las comprobaciones de autorización

Las comprobaciones de autorización se realizan cuando una aplicación intenta acceder a un gestor de colas, una cola, un proceso o una lista de nombres.

En IBM i, también se pueden realizar comprobaciones de autorización cuando un usuario emite un mandato de CL del Grupo 2 que accede a cualquiera de estos objetos de IBM MQ. Las comprobaciones se llevan a cabo en las siguientes circunstancias:

Cuando una aplicación se conecta a un gestor de colas utilizando una llamada MQCONN o MQCONNX

El gestor de colas solicita al entorno operativo el ID de usuario asociado a la aplicación. A continuación, el gestor de colas comprueba si el ID de usuario tiene autorización para conectarse al gestor de colas y retiene el ID de usuario para comprobaciones posteriores.

Los usuarios no tienen que iniciar la sesión en IBM MQ. IBM MQ presupone que los usuarios han iniciado la sesión en el sistema operativo subyacente y que éste los autentica.

Cuando una aplicación abre un objeto de IBM MQ utilizando una llamada MQOPEN o MQPUT1

Todas las comprobaciones de autorización se realizan cuando se abre un objeto y no cuando se accede al mismo posteriormente. Por ejemplo, las comprobaciones de autorización se realizan cuando una aplicación abre una cola. No se realizan cuando la aplicación coloca mensajes en la cola u los obtiene de ella.

Cuando una aplicación abre un objeto, especifica los tipos de operaciones que necesita realizar sobre el objeto. Por ejemplo, es posible que una aplicación abra una cola para explorar los mensajes que contiene y obtener los mensajes que contiene pero no para transferir mensajes a la cola. Para cada tipo de operación, el gestor de colas comprueba que el ID de usuario asociado a la aplicación tenga autorización para realizar esa operación

Cuando una aplicación abre una cola, las comprobaciones de autorización se realizan con respecto al objeto denominado en el campo `ObjectName` del descriptor de objeto. El campo `ObjectName` se utiliza en las llamadas `MQOPEN` o `MQPUT1`. Si el objeto es una cola de alias o una definición de cola remota, las comprobaciones de autorización se realizan respecto al propio objeto. No se realizan en la cola con la que se resuelven la cola de alias o la definición de la cola remota. Esto significa que el usuario no necesita tener permiso para acceder al mismo. Limite la autorización para crear colas a los usuarios con privilegios. De otro modo, los usuarios podrán eludir el control de accesos normal simplemente creando un alias.

Una aplicación puede hacer referencia a una cola remota de forma explícita. Establece los campos `ObjectName` y `ObjectQMgrName` en el descriptor de objeto en los nombres de la cola remota y el gestor de colas remoto. Las comprobaciones de autorización se realizan en la cola de transmisión con el mismo nombre que el gestor de colas remoto:

- z/OS En z/OS, se realiza una comprobación en el perfil de cola RACF que coincide con el nombre del gestor de colas remoto, y se realiza tanto si esta cola de transmisión está definida localmente como si no.
- Multi En Multiplatforms, se realiza una comprobación en el perfil RQMNAME que coincide con el nombre de gestor de colas remoto, si se utiliza la agrupación en clúster.

Una aplicación puede hacer referencia a una cola de clúster explícitamente estableciendo el campo `NombreObjeto` del descriptor de objeto en el nombre de la cola de clúster. Las comprobaciones de autorización se realizan sobre la cola de transmisión de clúster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

La autorización sobre una cola dinámica se basa en la cola modelo de la que se deriva, aunque no tiene por qué ser igual; consulte la nota 1.

El ID de usuario que el gestor de colas utiliza para las comprobaciones de autorización se obtiene del sistema operativo. El ID de usuario se obtiene cuando la aplicación se conecta al gestor de colas. Una aplicación con las autorizaciones adecuadas puede emitir una llamada `MQOPEN` especificando un ID de usuario alternativo. Las comprobaciones de control de accesos se realizan de este modo en el ID de usuario alternativo. Utilizar un ID de usuario alternativo no cambiará el ID de usuario asociado a la aplicación, solamente el que se utiliza para las comprobaciones de control de acceso.

Cuando una aplicación se suscribe a un tema utilizando una llamada MQSUB

Cuando una aplicación se suscribe a un tema, especifica el tipo de operación que necesita realizar. Está creando una suscripción o bien alterando una suscripción existente o bien reanudando una suscripción existente sin cambiarla. Para cada tipo de operación, el gestor de colas comprueba que el ID de usuario asociado a la aplicación tenga autorización para realizar la operación.

Cuando una aplicación se suscribe a un tema, las comprobaciones de autorización se realizan respecto a objetos de temas que se encuentran en el árbol de temas. Los objetos de temas están en el punto o encima del punto del árbol de temas al que se ha suscrito la aplicación. Las comprobaciones de autorización pueden implicar comprobaciones en más de un objeto de tema.El ID de usuario que el gestor de colas utiliza para las comprobaciones de autorización se obtiene del sistema operativo. El ID de usuario se obtiene cuando la aplicación se conecta al gestor de colas.

El gestor de colas realiza comprobaciones de autorización en las colas de suscriptores pero no en las colas gestionadas.

Cuando una aplicación suprime una cola dinámica persistente utilizando una llamada MQCLOSE
El manejador de objeto especificado en la llamada MQCLOSE no es necesariamente el mismo que el que ha devuelto la llamada MQOPEN que ha creado la cola dinámica persistente. Si es diferente, el gestor de colas comprueba el ID de usuario asociado a la aplicación que ha emitido la llamada MQCLOSE. Comprueba que el ID de usuario esté autorizado a suprimir la cola.

Cuando una aplicación que cierra una suscripción para eliminarla no la ha creado, se necesita la autorización de aplicación adecuada para eliminarla.

Cuando el servidor de mandatos procesa un mandato PCF que funciona en un objeto IBM MQ.
Esta regla incluye el caso en que un mandato PCF realiza una operación en un objeto de información de autenticación.

El ID de usuario que se utiliza para las comprobaciones de autorización es el que se ha encontrado en el campo UserIdentifier del descriptor de mensaje del mandato PCF. Este ID de usuario debe tener las autorizaciones necesarias sobre el gestor de colas en que se procesa el mandato. El mandato MQSC equivalente que se encapsula en un mandato PCF de escape se trata del mismo modo. Para obtener más información sobre el campo UserIdentifier y cómo establecerlo, consulte “Contexto de mensaje” en la página 101.

IBM i **En IBM i, cuando un usuario emite un mandato de CL del Grupo 2 que se ejecuta en un objeto de IBM MQ**
Esta regla incluye el caso en el que un mandato CL del Grupo 2 realiza una operación en un objeto de información de autenticación.

Las comprobaciones se realizan para determinar si el usuario tiene la autorización para realizar operaciones en un objeto de IBM MQ asociado al mandato. Las comprobaciones se realizan a menos que el usuario sea miembro del grupo QMQMADM o tenga autorización *ALLOBJ. La autorización necesaria depende del tipo de operación que el mandato realiza en el objeto. Por ejemplo, el mandato de **CHGMQMQ**, Cambiar cola MQM requiere la autorización para cambiar los atributos de la cola especificada por el mandato. Por el contrario, el mandato **DSPMQMQ**, Visualizar cola MQM requiere la autorización para visualizar los atributos de la cola especificada por el mandato.

Muchos mandatos se ejecutan en más de un objeto. Por ejemplo, para emitir el mandato **DLTMQMQ**, Suprimir cola MQM, se necesitan las autorizaciones siguientes:

- Autorización para conectar con el gestor de colas especificado en el mandato
- Autorización para suprimir la cola especificada en el mandato

Algunos mandatos no se ejecutan en ningún objeto. En este caso, el usuario sólo necesita autorización IBM i para emitir uno de estos mandatos. **STRMQMLSR** Iniciar escucha MQM, es un ejemplo de un mandato de este tipo.

Autoridad de usuario alternativo

Cuando una aplicación abre un objeto o se subscribe a un tema, la aplicación puede proporcionar un ID de usuario en la llamada MQOPEN, MQPUT1 o MQSUB. Puede solicitar al gestor de colas que utilice este ID de usuario para las comprobaciones de autorización, en lugar del ID asociado a la aplicación.

La aplicación solamente podrá abrir un objeto correctamente si se cumplen las dos condiciones siguientes:

- El ID de usuario asociado a la aplicación tiene autorización para proporcionar un ID de usuario diferente para las comprobaciones de autorización. Se considera que la aplicación tiene *autorización de usuario alternativo*.
- El ID de usuario que proporciona la aplicación tiene autorización para abrir el objeto para los tipos de operación solicitados o para suscribirse al tema.

Contexto de mensaje

La información del *contexto de mensaje* permite a la aplicación recuperar un mensaje para obtener información acerca de quién ha originado el mensaje. La información está contenida en campos del descriptor de mensaje y los campos se dividen en tres partes lógicas

Estas partes son las siguientes:

contexto de identidad
Estos campos contienen información acerca del usuario de la aplicación que ha transferido el mensaje a la cola.

contexto de origen
Estos campos contienen información acerca de la aplicación propiamente dicha y de cuándo se ha transferido el mensaje a la cola.

contexto de usuario
Estos campos contienen propiedades de mensaje que las aplicaciones pueden utilizar para seleccionar mensajes que el gestor de colas debe entregar.

Cuando una aplicación transfiere un mensaje a una cola, la aplicación puede solicitar al gestor de colas que genere información de contexto en el mensaje. Esta es la acción predeterminada. Alternativamente, puede especificar que los campos de contexto no contengan información. El ID de usuario asociado a una aplicación no requiere ninguna autorización especial para realizar estas acciones.

Una aplicación puede establecer los campos de contexto de identidad en un mensaje, lo que permite que el gestor de colas genere el contexto de origen, o puede establecer todos los campos de contexto. Una aplicación también puede pasar los campos de contexto de identidad de un mensaje que ha recuperado a un mensaje que va a transferir a una cola, o puede pasar todos los campos de contexto. Sin embargo, el ID de usuario asociado a una aplicación requiere autorización para establecer o pasar información de contexto. Una aplicación especifica que desea establecer o pasar información de contexto cuando abre la cola en la que está a punto de transferir los mensajes y es en dicho momento cuando se comprueba su autorización.

La siguiente es una breve descripción de cada uno de los campos de contexto:

Contexto de identidad

UserIdentifier
El ID de usuario asociado a la aplicación que ha transferido el mensaje. Si el gestor de colas establece este campo, se establecerá en el ID de usuario que se obtiene del sistema operativo cuando la aplicación se conecta al gestor de colas.

AccountingToken
La información que se puede utilizar para cobrar por el trabajo realizado como resultado de un mensaje.

ApplIdentityData
Si el ID de usuario asociado a una aplicación tiene autorización para establecer los campos de contexto de identidad o para establecer todos los campos de contexto, la aplicación puede establecer este campo en cualquier valor relacionado con la identidad. Si el gestor de colas establece este campo, se establece en blanco.

Contexto de origen

PutApplType
El tipo de la aplicación que ha transferido el mensaje; por ejemplo, una transacción CICS.

PutApplName
El nombre de la aplicación que ha transferido el mensaje.

PutDate
La fecha en que se ha transferido el mensaje.

PutTime
La hora en la que se ha transferido el mensaje.

ApplOriginData
Si el ID de usuario asociado a una aplicación tiene autorización para establecer todos los campos de contexto, la aplicación puede establecer este campo en cualquier valor relacionado con el origen. Si el gestor de colas establece este campo, se establece en blanco.

Contexto de usuario

Los valores siguientes están soportados para **MQINQMP** o **MQSETMP**:

MQPD_USER_CONTEXT

La propiedad está asociada al contexto de usuario.

No se requiere ninguna autorización especial para poder establecer una propiedad asociada al contexto de usuario utilizando la llamada MQSETMP.

En un gestor de colas de la versión 7.0 o posterior, una propiedad asociada al contexto de usuario se guarda tal como se describe para MQOO_SAVE_ALL_CONTEXT. Una llamada MQPUT con MQOO_PASS_ALL_CONTEXT especificado hace que la propiedad se copie del contexto guardado al nuevo mensaje.

MQPD_NO_CONTEXT

La propiedad no está asociada a un contexto de mensaje.

Un valor no reconocido se rechaza con MQRC_PD_ERROR. El valor inicial de este campo es **MQPD_NO_CONTEXT**.

Para obtener una descripción detallada de cada uno de los campos de contexto, consulte MQMD - Descriptor de mensaje. Para obtener más información acerca de cómo utilizar el contexto de mensaje, consulte Contexto de mensaje.

IBM i ALW *Autorización para trabajar con objetos IBM MQ en sistemas* IBM i *IBM i , AIX, Linux, and Windows*

El componente de servicio de autorización suministrado con IBM MQ se denomina *gestor de autorizaciones sobre objetos* (OAM). Proporciona control de acceso a través de comprobaciones de autenticación y autorización.

AUTENTICACIÓN.
La comprobación de la autenticación ejecutada por el OAM que se suministra con IBM MQ es básica y sólo se realiza en circunstancias específicas. No está diseñada para cumplir con los requisitos estrictos previstos en un entorno muy seguro.

El OAM realiza su comprobación de autenticación cuando una aplicación se conecta a un gestor de colas y se cumplen las condiciones siguientes:

- Si la aplicación de conexión ha proporcionado una estructura MQCSP, y
- Al atributo *AuthenticationType* de la estructura MQCSP se le proporciona el valor MQCSP_AUTH_USER_ID_AND_PWD y
- El valor CHCKLOCL o CHKCCLNT en el objeto AUTHINFO configurado no es 'NONE'

Los pasos de autenticación en el OAM validan la contraseña utilizando los servicios del sistema operativo, que pueden haberse configurado para realizar comprobaciones adicionales, como por ejemplo asegurarse de que el nombre de usuario no ha tenido demasiados intentos de prueba de contraseña incorrectos.

Es posible utilizar mecanismos de autenticación alternativos si escribe un nuevo componente de servicio de autorización u obtiene uno de un proveedor.

La autorización.

Las comprobaciones de la autorización son exhaustivas y no están diseñadas para cumplir la mayoría de requisitos normales.

Las comprobaciones de autorización se realizan cuando una aplicación emite una llamada MQI para acceder a un gestor de colas, una cola, un proceso o una lista de nombres. También se realizan en otros momentos; por ejemplo, cuando un mandato está siendo ejecutado por el servidor de mandatos.

En los sistemas IBM i IBM i , AIX, Linux, and Windows, el *servicio de autorización* proporciona el control de accesos cuando una aplicación emite una llamada MQI para acceder a un objeto de IBM MQ que es un gestor de colas, cola, proceso, tema o lista de nombres. Esto incluye comprobaciones de la autorización de usuario alternativo y la autorización para establecer o pasar información de contexto.

Windows En Windows, el OAM otorga a los miembros del grupo Administradores la autorización para acceder a todos los objetos de IBM MQ, aunque el UAC esté habilitado. Además, en sistemas Windows , la cuenta SYSTEM tiene acceso completo a los recursos de IBM MQ .

El servicio de autorización también proporciona comprobaciones de autorización cuando un mandato PCF realiza operaciones en uno de estos objetos de IBM MQ o en un objeto de información de autenticación. El mandato MQSC equivalente que se encapsula en un mandato PCF de escape se trata del mismo modo.

IBM i En IBM i, a menos que el usuario sea miembro del grupo QMQMADM o tenga la autorización *ALLOBJ, el servicio de autorización también proporciona comprobaciones de autorización cuando un usuario emite un mandato de CL del Grupo 2 que se ejecuta en cualquiera de estos objetos de IBM MQ o en un objeto de información de autenticación.

El servicio de autorización es un *servicio instalable*, lo que significa que lo implementan uno o varios *componentes de servicio instalables*. Todo componente se invoca mediante una interfaz documentada. Esto permite que los usuarios y proveedores suministren componentes que mejoran o sustituyen los que se proporcionan con los productos de IBM MQ.

El componente de servicio de autorización suministrado con IBM MQ se denomina gestor de autorizaciones sobre objetos (OAM). El OAM se habilita automáticamente para cada gestor de colas que cree.

El OAM mantiene una lista de control de accesos (ACL) para cada objeto de IBM MQ al que controla el acceso. En los sistemas AIX and Linux, sólo los ID de grupo pueden aparecer en una ACL. Esto significa que todos los miembros de un grupo tienen las mismas autorizaciones. En los sistemas IBM i IBM i y Windows, ambos ID de usuario e ID de grupo pueden aparecer en una ACL. Esto significa que se pueden otorgar autorizaciones a usuarios y grupos individuales.

Se aplica una limitación de 12 caracteres al ID de grupo y de usuario. Las plataformas UNIX suelen restringir la longitud de un ID de usuario a 12 caracteres. AIX y Linux han elevado este límite pero IBM MQ continúa cumpliendo una restricción de 12 caracteres en todas las plataformas UNIX. Si utiliza un ID de usuario de más de 12 caracteres, IBM MQ lo sustituye por el valor "UNKNOWN". No defina un ID de usuario con un valor de "UNKNOWN".

El gestor de autorizaciones sobre objetos puede autenticar un usuario y cambiar los campos de contexto de identidad apropiados. Se habilita especificando una estructura de parámetros de seguridad (MQCSP) en una llamada MQCONNX. La estructura se pasa a la función Autenticar un usuario del gestor de autorizaciones (MQZ_AUTHENTICATE_USER), que establece los campos de contexto de identidad apropiados. Si es una conexión MQCONNX desde un cliente IBM MQ, la información de MQCSP se transmite al gestor de colas al que el cliente se conecta a través de la conexión con el cliente y el canal de conexión con el servidor. Si las salidas de seguridad se definen en dicho canal, el MQCSP se pasa a cada salida de seguridad y puede ser alterado por la salida. Las salidas de seguridad también pueden crear el MQCSP. Para obtener más detalles sobre el uso de las salidas de seguridad en este contexto, consulte Programas de salida de seguridad de canal.

Aviso: En algunos casos, la contraseña en la estructura MQCSP para una aplicación cliente se enviará por una red en texto sin formato. Para asegurarse de que las contraseñas de la aplicación cliente están protegidas adecuadamente, consulte IBM MQProtección de contraseña de CSP.

En sistemas AIX, Linux, and Windows , el mandato de control **setmqaut** otorga y revoca autorizaciones y se utiliza para mantener las ACL. Por ejemplo, el mandato:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER +browse +get
```

permite que los miembros del grupo VOYAGER exploren los mensajes de la cola MOON.EUROPA propiedad del gestor de colas JUPITER. También permite que los miembros obtengan mensajes de la cola. Para revocar estas autorizaciones posteriormente, especifique el siguiente mandato:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER -browse -get
```

El mandato:

```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +put
```

permite que los miembros del grupo VOYAGER transfieran mensajes a cualquier cola cuyo nombre empiece por los caracteres `MOON.`. MOON.* es el nombre de un perfil genérico.Un *perfil genérico* le permite otorgar autorizaciones para un conjunto de objetos utilizando un único mandato **setmqaut** .

El mandato de control **dspmqaut** está disponible para visualizar las autorizaciones actuales que un usuario o un grupo tiene para un objeto especificado.El mandato de control **dmpmqaut** también está disponible para visualizar las autorizaciones actuales asociadas con los perfiles genéricos.

IBM i En IBM i, un administrador utiliza el mandato de CL GRTMQMAUT para otorgar autorizaciones y el mandato de CL RVKMQMAUT para revocarlas. También se pueden utilizar perfiles genéricos.Por ejemplo, el mandato de CL:

```
GRTMQMAUT MQMNAME(JUPITER) OBJTYPE(*Q) OBJ('MOON.*') USER(VOYAGER) AUT(*PUT)
```

proporciona la misma función que el ejemplo anterior de un mandato **setmqaut**; permite que los miembros del grupo VOYAGER transfieran mensajes a cualquier cola cuyo nombre empiece por los caracteres `MOON.`

IBM i El mandato de CL DSPMQMAUT visualiza las autorizaciones actuales que un usuario o un grupo tiene para un objeto especificado. Los mandatos de CL WRKMQMAUT y WRKMQMAUTD también están disponibles para trabajar con las autorizaciones actuales asociadas a objetos y perfiles genéricos.

Si no desea realizar comprobaciones de seguridad como sería el caso, por ejemplo, de un entorno de prueba, puede inhabilitar el OAM.

Multi *Utilización de PCF para acceder a los mandatos del gestor de autorizaciones sobre objetos (OAM)*
En sistemas IBM i, AIX, Linux, and Windows, puede utilizar mandatos PCF para acceder a mandatos de administración de OAM.

Los mandatos PCF y los mandatos gestor de autorizaciones sobre objetos equivalentes son los siguientes:

Tabla 8. Mandatos PCF y los mandatos gestor de autorizaciones sobre objetos

mandato PCF	**mandatos gestor de autorizaciones sobre objetos**
Consultar registros de autorización	dmpmqaut
Consultar entidad de autorización	dspmqaut
Establecer registro de autorización	setmqaut

Tabla 8. Mandatos PCF y los mandatos gestor de autorizaciones sobre objetos (continuación)	
mandato PCF	**mandatos gestor de autorizaciones sobre objetos**
Suprimir registro de autorización	setmqaut con -opción de eliminación

Los mandatos **setmqaut** y **dmpmqaut** están restringidos a los miembros del grupo mqm. Los mandatos PCF equivalentes los pueden ejecutar usuarios de cualquier grupo a los que se les haya otorgado autorizaciones dsp y chg en el gestor de colas.

Si desea más información sobre cómo utilizar estos mandatos, consulte Introducción a formatos de mandato programables.

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.

Seguridad de la mensajería remota

En este apartado se tratan aspectos relativos a la seguridad de la mensajería remota.

Debe proporcionar autorización a los usuarios para que utilicen los recursos de IBM MQ. Esto se organiza de acuerdo con las acciones que se han de tomar con respecto a los objetos y definiciones. Por ejemplo:

- Los usuarios autorizados son los que pueden iniciar y detener los gestores de colas
- Las aplicaciones deben conectarse con el gestor de colas y tener autorización para utilizar colas
- Los usuarios autorizados deben crear y controlar los canales de mensajes
- Los objetos se mantienen en las bibliotecas y el acceso a éstas puede restringirse

El agente de canal de mensajes en un sitio remoto debe comprobar que el mensaje que se entregan se ha originado desde un usuario con autorización para hacerlo en este sitio remoto. Además, dado que los MCA se pueden iniciar de forma remota, es posible que sea necesario verificar que los procesos remotos que están intentando iniciar sus MCA estén autorizados a hacerlo. Hay cuatro posibles formas de hacerlo:

1. Utilice adecuadamente el atributo PutAuthority de la definición de canal RCVR, RQSTR o CLUSRCVR para controlar qué usuario se utiliza para las comprobaciones de autorización cuando los mensajes entrantes se colocan en las colas. Consulte la descripción del mandato DEFINE CHANNEL en la Consulta de mandatos MQSC.
2. Implemente registros de autenticación de canal para rechazar los intentos de conexión no deseados o para establecer un valor MCAUSER basada en lo siguiente: la dirección IP remota, el ID de usuario remoto, el Nombre distinguido del asunto (DN) TLS proporcionado o el nombre del gestor de colas remoto.
3. Implemente la comprobación de seguridad de la *salida de usuario* para asegurarse de que el canal de mensajes correspondiente está autorizado. La seguridad de la instalación que alberga el canal correspondiente asegura que todos los usuarios están debidamente autorizados, por lo que no es necesario comprobar los mensajes individuales.
4. Implemente el proceso de mensajes de la *salida de usuario* para asegurarse de que se comprueba la autorización de los mensajes individuales.

IBM i *Seguridad de objetos de IBM MQ for IBM i*

En este apartado se tratan aspectos relativos a la seguridad de la mensajería remota.

Debe proporcionar autorización a los usuarios para que hagan uso de los recursos de IBM MQ for IBM i. Esta autorización está organizada según las acciones que se emprenderán respecto a objetos y definiciones. Por ejemplo:

- Los usuarios autorizados son los que pueden iniciar y detener los gestores de colas
- Las aplicaciones deben conectarse al gestor de colas y tener autorización para hacer uso de las colas
- Usuarios autorizados deben crear y controlar los canales de mensajes

El agente de canal en un sitio remoto debe comprobar si el mensaje que se entrega ha derivado de un usuario con autorización para emitir el mensaje en este sitio remoto. Además, dado que los MCA se pueden iniciar de forma remota, es posible que sea necesario verificar que los procesos remotos que están intentando iniciar sus MCA estén autorizados a hacerlo. Hay cuatro posibles formas de hacerlo:

- Decrete en la definición de canal que los mensajes deben contener autorización de *contexto* aceptable; de lo contrario quedan descartados.
- Implemente registros de autorización de canal para rechazar intentos de conexión no deseados, o bien para establecer un valor MCAUSER basado en uno de los siguientes: la dirección IP remota, el ID de usuario remoto, el nombre distinguido (DN) de TLS proporcionado o el nombre del gestor de colas remoto.
- Implemente la comprobación de seguridad de salida del usuario para garantizar que el canal de mensajes correspondiente está autorizado. La seguridad de la instalación que alberga el canal correspondiente asegura que todos los usuarios están debidamente autorizados, por lo que no es necesario comprobar los mensajes individuales.
- Implemente el proceso de mensajes de salida del usuario para garantizar que los mensajes individuales se sometan a examen para su autorización.

Estos son algunos hechos sobre el modo en que IBM MQ for IBM i opera la seguridad:

- IBM i identifica y autentica a los usuarios.
- Los servicios del gestor de colas invocadas por aplicaciones se ejecutan con la autorización del perfil de usuario del gestor de colas, pero en el proceso del usuario.
- Los servicios del gestor de colas invocados mediante mandatos de usuario se ejecutan con la autorización del perfil de usuario del gestor de colas.

Linux AIX *Seguridad de objetos en AIX and Linux*

Los usuarios de administración deben formar parte del grupo mqm en el sistema (incluido raíz) si este ID va a utilizar mandatos de administración de IBM MQ.

Siempre debe ejecutar amqcrsta con el ID de usuario "mqm".

ID de usuario en AIX and Linux

El gestor de colas convierte todos los identificadores de usuario en mayúsculas o en una combinación de mayúsculas/minúsculas en minúsculas. A continuación, el gestor de colas inserta los identificadores de usuario en la parte de contexto de un mensaje o comprueba su autorización. Por consiguiente, las autorizaciones sólo están basadas en identificadores en minúsculas.

Windows *Seguridad de objetos en sistemas Windows*

Los usuarios de administración deben formar parte del grupo mqm y del grupo de administradores en sistemas Windows si este ID va a utilizar mandatos de administración de IBM MQ.

ID de usuario en sistemas Windows

En sistemas Windows, *si no hay ninguna salida de mensaje instalada*, el gestor de colas convierte los identificadores en mayúsculas o en una combinación de mayúsculas/minúsculas en minúsculas. A continuación, el gestor de colas inserta los identificadores de usuario en la parte de contexto de un mensaje o comprueba su autorización. Por consiguiente, las autorizaciones sólo están basadas en identificadores en minúsculas.

ID de usuario en sistemas

Las plataformas que no sean sistemas AIX, Linux, and Windows utilizan caracteres en mayúsculas para los ID de usuario en los mensajes. Para permitir que los sistemas AIX, Linux, and Windows utilicen ID de usuario en minúsculas en los mensajes, el agente de canal de mensajes (MCA) debe llevar a cabo las conversiones adecuadas de caracteres alfabéticos.

Para permitir que los sistemas AIX, Linux, and Windows utilicen los ID de usuario en minúsculas en los mensajes, el agente de canal de mensajes (MCA) lleva a cabo las conversiones siguientes en estas plataformas:

En el extremo emisor
Los caracteres alfabéticos en todos los ID de usuario se convierten en caracteres en mayúsculas, si no hay ningún mensaje de salida instalado.

En el extremo receptor
Los caracteres alfabéticos en todos los ID de usuario se convierten en caracteres en minúsculas, si no hay ningún mensaje de salida instalado.

Las conversiones automáticas no se realizan si proporciona una salida de mensaje en AIX, Linux, and Windows por cualquier otro motivo.

Utilización de un servicio de autorización personalizado

IBM MQ proporciona un servicio de autorización instalable. Puede optar por instalar un servicio alternativo.

El componente del servicio de autorización proporcionado con IBM MQ se llama Gestor de autoridad de objeto (OAM). Si el OAM no proporciona los recursos de autorización que necesita, puede escribir su propio componente de servicio de autorización. Las funciones de servicio instalables que debe implementar un componente de servicio de autorización se describen en Información de consulta de la interfaz de servicios instalables.

Control de accesos para clientes

El control de accesos se basa en los ID de usuario. Puede haber muchos ID de usuario para administrar, y pueden estar en distintos formatos. Puede establecer la propiedad de canal de conexión del servidor MCAUSER en un valor de ID de usuario especial para que puedan utilizarla los clientes.

El control de acceso en IBM MQ está basado en los ID de usuario. Normalmente se utiliza ID de usuario del proceso que realiza llamadas MQI. En el caso de clientes MQI de MQ, el MCA de conexión con el servidor efectúa llamadas MQI en nombre de clientes MQI de MQ. Puede seleccionar un ID de usuario alternativo para que lo utilice el MCA de conexión con el servidor para efectuar llamadas MQI. El ID de usuario alternativo se puede asociar con la estación de trabajo del cliente o lo que elija para organizar y controlar el acceso de los clientes. El ID de usuario debe tener las autorizaciones necesarias asignadas a éste en el servidor para emitir llamadas MQI. Elegir un ID de usuario alternativo es preferible a permitir que los clientes efectúen llamadas MQI con la autorización del MCA de conexión con el cliente.

Tabla 9. El ID de usuario utilizado por un canal de conexión del servidor

ID de usuario	Cuándo se utiliza
ID de usuario establecido por una salida de seguridad	Se utiliza a menos que lo bloquee una regla **CHLAUTH TYPE(BLOCKUSER)**. Consulte la sección siguiente, “Establecimiento del ID de usuario en una salida de seguridad” en la página 109, si desea más información.
ID de usuario establecido por una regla CHLAUTH	Se utiliza a menos que una salida de seguridad lo sobrescriba. Consulte Registros de autenticación de canal para obtener más información.
ID de usuario definido en el atributo **MCAUSER** en la definición de canal SVRCONN	Se utiliza a menos que una salida de seguridad o una regla CHLAUTH lo sobrescriban.
ID de usuario que fluye desde la máquina cliente	Se utiliza cuando no se haya establecido ningún ID de usuario de cualquier otro modo.
ID de usuario que ha iniciado el canal de conexión del servidor	Se utiliza si no se ha establecido ningún ID de usuario de ningún otro modo y no se ha producido un flujo de ID de usuario de cliente. Consulte la sección siguiente, “El ID de usuario que ejecuta el programa de canal” en la página 109, si desea más información.

Puesto que el MCA de conexión con el servidor efectúa llamadas MQI en nombre de usuarios remotos, es importante tener en cuenta las implicaciones de seguridad del MCA de conexión con el servidor que emite llamadas MQI en nombre de clientes remotos y cómo administrar el acceso de un gran número potencial de usuarios.

- Una alternativa es que el MCA de conexión con el servidor emita llamadas MQI con su propia autorización. Pero tenga en cuenta que, no es deseable generalmente que el MCA de conexión con el servidor, con sus potentes prestaciones de acceso, emita llamadas MQI en nombre de usuarios de cliente.
- Otra alternativa es utilizar el ID de usuario que fluye del cliente. El MCA de conexión con el servidor emite llamadas MQI utilizando las prestaciones de acceso del ID de usuario del cliente. Este enfoque presenta una serie de preguntas que hay que tener en cuenta:
  1. Existen diferentes formatos para el ID de usuario en diferentes plataformas. Esto a veces provoca problemas si el formato del ID de usuario en el cliente difiere de los formatos aceptables en el servidor.
  2. Existen potencialmente muchos clientes, con ID de usuario diferentes y cambiantes. Los ID deben definirse y gestionarse en el servidor.
  3. ¿Es el ID de usuario fiable? Cualquier ID de usuario puede transmitirse de un cliente, no necesariamente el ID del usuario registrado. Por ejemplo, el cliente puede transmitir un ID con plena autorización `mqm` que intencionadamente sólo estaba definido en el servidor por razones de seguridad.
- La alternativa preferida es definir las señales de identificación del cliente en el servidor y limitar así las posibilidades de las aplicaciones conectadas del cliente. Esto se suele realizar estableciendo la propiedad de canal de conexión con el servidor MCAUSER en un valor de ID de usuario especial que utilizarán los clientes y definiendo unos pocos ID para que los utilicen los clientes con diferente nivel de autorización en el servidor.

Establecimiento del ID de usuario en una salida de seguridad

Para IBM MQ MQI clients, el proceso que emite las llamadas MQI es el MCA de conexión con el servidor. El ID de usuario que utiliza el MCA de conexión con el servidor está contenido en los campos `MCAUserIdentifier` o `LongMCAUserIdentifier` del MQCD. El contenido de estos campos viene establecido por:

- Cualquier valor definido por las rutinas de salida de seguridad
- El ID de usuario del cliente
- MCAUSER (en la definición de canal de conexión con el servidor)

La salida de seguridad puede prevalecer sobre los valores que son visibles, cuando se invoca.

- Si el atributo MCAUSER del canal de conexión con el servidor no está establecido en blanco, se utiliza el valor MCAUSER.
- Si el atributo MCAUSER del canal de conexión con el servidor está en blanco, se utiliza el ID de usuario procedente del cliente.
- Si el atributo MCAUSER del canal de conexión con el cliente está en blanco y no se recibe ningún ID de usuario del cliente, se utiliza el ID de usuario que inició el canal de conexión con el servidor.

El cliente IBM MQ no fluye el ID de usuario declarado hasta el servidor cuando se está utilizando una salida de seguridad del lado del cliente.

El ID de usuario que ejecuta el programa de canal

Cuando los campos de ID de usuario se derivan del ID de usuario que inició el canal de conexión con el servidor, se utiliza el valor siguiente:

- z/OS Para z/OS, el ID de usuario asignado a la tarea iniciada de iniciador de canal mediante la tabla de procedimientos iniciados de z/OS.

- Para TCP/IP (no z/OS), el ID de usuario de la entrada `inetd.conf` o el ID de usuario que ha iniciado el escucha.
- Para SNA (no z/OS), el ID de usuario de la entrada de servidor SNA o (si no hay ninguno) la solicitud de conexión entrante o el ID de usuario que ha iniciado el escucha.
- Para NetBIOS o SPX, el ID de usuario que ha iniciado el escucha.

Si existe alguna definición de canal de conexión con el servidor cuyo atributo MCAUSER esté en blanco, los clientes pueden utilizar esa definición de canal para conectarse con el gestor de colas con una autorización de acceso determinada por el ID de usuario suministrado por el cliente. Esto puede representar un riesgo para la seguridad si el sistema en el que se ejecuta el gestor de colas permite conexiones no autorizadas a la red. El canal de conexión de servidor predeterminado de IBM MQ (SYSTEM.DEF.SVRCONN) tiene el atributo MCAUSER establecido en blanco. Para impedir el acceso no autorizado, actualice el atributo MCAUSER de la definición predeterminada con un ID de usuario que no tenga acceso a los objetos de IBM MQ MQ.

El caso de los ID de usuarios

Cuando defina un canal con `runmqsc`, el atributo MCAUSER quedará en mayúsculas a menos que el ID de usuario esté entre comillas simples.

ALW En servidores de AIX, Linux, and Windows, el contenido del campo `MCAUserIdentifier` que se recibe del cliente cambia a minúsculas.

IBM i En servidores de IBM i, el contenido del campo `LongMCAUserIdentifier` que se recibe del cliente cambia a mayúsculas.

Linux AIX En servidores en sistemas AIX and Linux, el contenido del campo `LongMCAUserIdentifier` que se recibe del cliente cambia a minúsculas.

De forma predeterminada, el ID de usuario que se pasa cuando se utiliza una aplicación de enlace IBM MQ JMS, es el ID de usuario para la JVM en la que se está ejecutando la aplicación.

También es posible pasar un ID de usuario a través del método `createQueueConnection`.

Planificación de la confidencialidad

Debe planificar mantener los datos confidenciales.

Puede implementar la confidencialidad a nivel de aplicación o a nivel de enlace. Puede elegir utilizar TLS, en cuyo caso debe planificar el uso de certificados digitales. También puede utilizar programas de salida de canal si los recursos estándares no satisfacen los requisitos.

Conceptos relacionados

“Comparación entre la seguridad a nivel de enlace y la seguridad a nivel de aplicación” en la página 110
Este tema contiene información sobre distintos aspectos de la seguridad de nivel de enlace y de nivel de aplicación y compara los dos niveles de seguridad.

“Programas de salida de canal” en la página 116
Los *programas de salida de canal* son programas a los que se llama en lugares definidos de la secuencia de proceso de un MCA. Los usuarios y proveedores pueden escribir sus propios programas de salida de canal. IBM proporciona algunos de ellos.

“Protección de canales con SSL/TLS” en la página 122
El soporte de TLS en IBM MQ utiliza el objeto de información de autenticación del gestor de colas y diversos mandatos MQSC. También debe contemplar el uso de certificados digitales.

Comparación entre la seguridad a nivel de enlace y la seguridad a nivel de aplicación

Este tema contiene información sobre distintos aspectos de la seguridad de nivel de enlace y de nivel de aplicación y compara los dos niveles de seguridad.

La seguridad a nivel de enlace y la seguridad a nivel de aplicación se ilustran en la Figura 10 en la página 111.

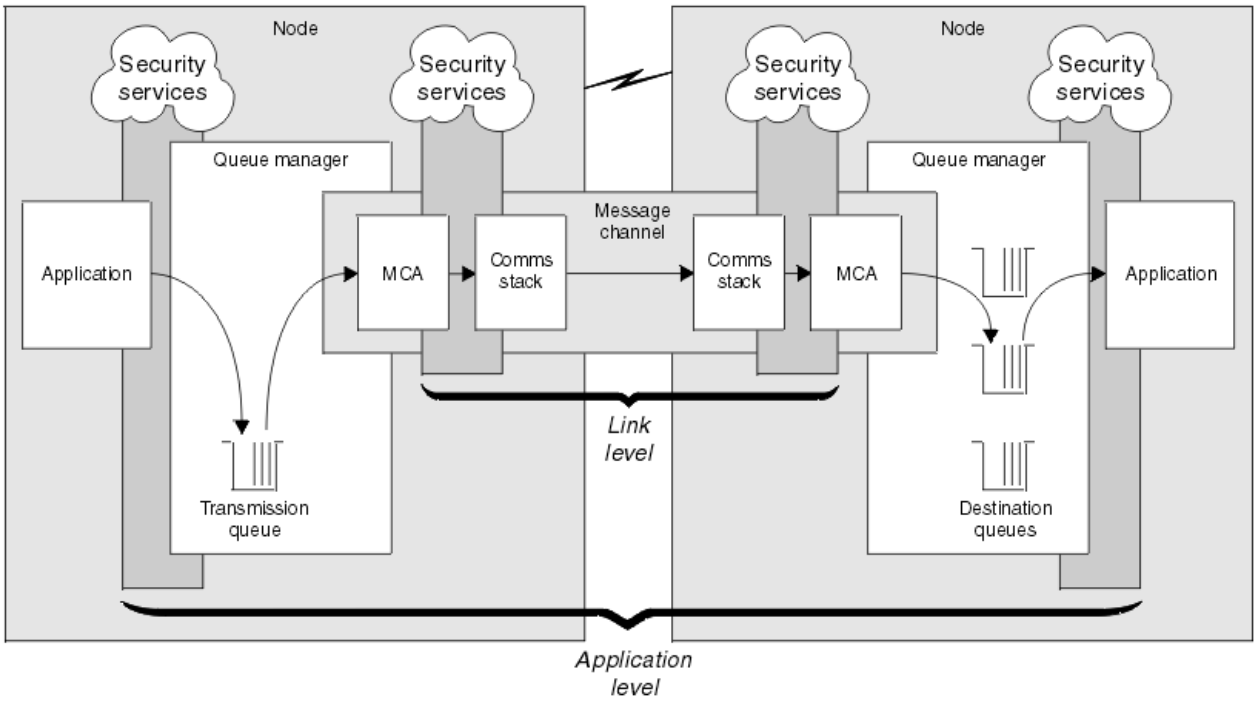


Figura 10. Seguridad a nivel de enlace y seguridad a nivel de aplicación

Protección de los mensajes en las colas

La seguridad a nivel de enlace puede proteger los mensajes mientras se transfieren de un gestor de colas a otro. Esto es especialmente importante cuando los mensajes se transmiten a través de una red que no es segura. No obstante, no puede proteger los mensajes mientras están almacenados en las colas de un gestor de colas de origen, de un gestor de colas de destino o de un gestor de colas intermedio.

z/OS El cifrado de conjuntos de datos de z/OS puede proporcionar cierta protección de los mensajes almacenados en las colas, pero sólo para los datos que se encuentran en reposo en un gestor de colas local. Consulte la sección Confidencialidad para los datos en reposo en IBM MQ for z/OS con cifrado de conjunto de datos. para obtener más información.

La seguridad a nivel de aplicación puede, en comparación, proteger los mensajes cuando están almacenados en colas y se aplica incluso cuando no se utiliza la gestión de colas distribuida. Esta es la diferencia más importante entre la seguridad a nivel de enlace y la seguridad a nivel de aplicación que se ilustra en la Figura 10 en la página 111.

Gestores de colas que no se ejecutan en entornos controlados y fiables

Si un gestor de colas se está ejecutando en un entorno controlado y de confianza, los mecanismos de control de accesos que proporciona IBM MQ pueden considerarse suficientes para proteger los mensajes almacenados en las colas. Esto es especialmente cierto si solamente se trata de gestionar colas locales y los mensajes no salen nunca del gestor de colas. La seguridad a nivel de aplicación en este caso puede considerarse innecesaria.

La seguridad a nivel de aplicación también puede considerarse innecesaria si los mensajes se transfieren a otro gestor de colas que también está ejecutándose en un entorno controlado y fiable o si se reciben desde un gestor de colas de este tipo. La necesidad de seguridad a nivel de aplicación aumenta cuando los mensajes se transfieren a o se reciben de un gestor de colas que no está ejecutándose en un entorno controlado y fiable.

Diferencias de coste

La seguridad a nivel de aplicación puede costar más que la seguridad a nivel de enlace en lo que se refiere a la administración y el rendimiento.

Es probable que el coste de la administración sea mayor porque potencialmente hay más restricciones a la hora de configurar y mantener. Por ejemplo, es posible que deba asegurarse de que un usuario determinado envía solamente determinados tipos de mensajes y envía mensajes solamente a determinados destinos. Por el contrario, es posible que tenga que asegurarse de que un usuario determinado recibe solamente determinados tipos de mensajes y únicamente de determinadas fuentes. En lugar de gestionar los servicios de seguridad a nivel de enlace en un solo canal de mensajes, es posible que tenga que configurar y mantener reglas para cada par de usuarios que intercambie mensajes a través de dicho canal.

El rendimiento puede verse afectado si los servicios de seguridad se invocan cada vez que una aplicación transfiere u obtiene un mensaje.

Las organizaciones tienden a considerar en primer lugar la seguridad a nivel de enlace porque resulta más fácil de implementar. La seguridad a nivel de aplicación la tienen en cuenta si descubren que la seguridad a nivel de enlace no satisface todos sus requisitos.

Disponibilidad de los componentes

Generalmente, en un entorno distribuido, un servicio de seguridad requiere un componente en al menos dos sistemas. Por ejemplo, es posible que un mensaje se cifre en un sistema y se descifre en otro. Esto se aplica a la seguridad a nivel de enlace y a la seguridad a nivel de aplicación.

En un entorno heterogéneo en el que se utilicen diferentes plataformas y cada una de las cuales tenga diferentes niveles de funciones de seguridad, es posible que los componentes necesarios de un servicio de seguridad no estén disponibles para cada plataforma en la que se necesitan y con un formato que resulte fácil de utilizar. Probablemente, esto se deba tener más en cuenta en la seguridad a nivel de aplicación que en la seguridad a nivel de enlace, sobre todo si piensa proporcionar su propio nivel de seguridad a nivel de aplicación comprando componentes de fuentes diferentes.

Mensajes de la cola de mensajes no entregados

Si un mensaje está protegido por la seguridad a nivel de aplicación, es posible que exista algún problema si, por algún motivo, el mensaje no llega a su destino y se coloca en una cola de mensajes no entregados. Si no encuentra el modo de procesar el mensaje a partir de la información incluida en el descriptor de mensaje y la cabecera de la cola de mensajes no entregados, es posible que tenga que examinar el contenido de los datos de la aplicación. Esto no podrá hacerlo si los datos de la aplicación están cifrados y el único que puede descifrarlos es el destinatario.

Funciones que no puede realizar la seguridad a nivel de aplicación

La seguridad a nivel de aplicación no es una solución completa. Incluso si implementa la seguridad a nivel de aplicación, es posible que necesite algunos servicios de seguridad a nivel de enlace. Por ejemplo:

- Cuando se inicia un canal, la autenticación mutua de los dos MCA puede seguir siendo un requisito. Esto solamente puede llevarlo a cabo mediante el servicio de seguridad a nivel de enlace.
- La seguridad a nivel de aplicación no puede proteger la cabecera de la cola de transmisión, MQXQH, que incluye el descriptor de mensaje intercalado. Ni tampoco puede proteger los datos de los flujos de protocolo de canal de IBM MQ que no sean los datos de mensaje. Solamente la seguridad de enlace puede proporcionar esta protección.
- Si se invocan los servicios de seguridad a nivel de aplicación en el extremo del servidor de un canal MQI, los servicios no pueden proteger los parámetros de las llamadas MQI que se envían a través del canal. En especial, los datos de la aplicación de una llamada MQPUT, MQPUT1 o MQGET no están protegidos. Solamente la seguridad a nivel de enlace puede proporcionar protección en este caso.

Seguridad a nivel de enlace

La *seguridad a nivel de enlace* hace referencia a los servicios de seguridad que invoca, de forma directa o indirecta, un MCA, el subsistema de comunicaciones o una combinación de ambos que funcionen conjuntamente.

La seguridad a nivel de enlace se ilustra en la Figura 10 en la página 111.

Los siguientes son ejemplos de servicios de seguridad a nivel de enlace:

- El MCA a cada extremo de un canal de mensajes puede autenticar a su asociado. Esto se lleva a cabo cuando se inicia el canal y se establece una conexión de comunicaciones pero antes de que se inicie el flujo de los mensajes. Si la autenticación no se ejecuta correctamente en alguno de los extremos, el canal se cierra y no se transfiere ningún mensaje. Este es un ejemplo de un servicio de identificación y autenticación.
- Se puede cifrar un mensaje en el extremo emisor de un canal y descifrar en el extremo receptor. Este es un ejemplo de un servicio de confidencialidad.
- Un mensaje se puede comprobar en el extremo receptor de un canal para determinar si el contenido se ha modificado de forma deliberada mientras se estaba transmitiendo a través de la red. Este es un ejemplo de un servicio de integridad de datos.

Seguridad a nivel de enlace proporcionada por IBM MQ

El principal medio de provisión de confidencialidad e integridad de datos en IBM MQ es mediante el uso de TLS. Para obtener más información sobre el uso de TLS en IBM MQ, consulte “Protocolos de seguridad TLS en IBM MQ” en la página 25. Para la autenticación, IBM MQ proporciona el recurso para utilizar registros de autenticación de canal. Los registros de autenticación de canal ofrecen un control preciso sobre el acceso otorgado a los sistemas que se conectan, a nivel de canales individuales o de grupos de canales. Para obtener más información, consulte “Registros de autenticación de canal” en la página 53.

Cómo proporcionar su propia seguridad a nivel de enlace

Puede proporcionar sus propios servicios de seguridad de nivel de enlace. Escribir sus propios programas de salida de canal es el método principal para proporcionar sus propios servicios de seguridad a nivel de enlace.

Se proporciona una introducción a los programas de salida de canal en “Programas de salida de canal” en la página 116. El mismo tema también describe el programa de salida de canal que se proporciona con IBM MQ for Windows (el programa de salida de canal SSPI). Este programa de salida de canal se suministra en formato fuente para que pueda modificar el código fuente para ajustarlo a sus necesidades. Si este programa de salida de canal, o los programas de salida de canal disponibles de otros proveedores, no se ajustan a sus requisitos, puede diseñar y escribir el suyo propio. En este tema se sugieren formas en que los programas de salida de canal pueden proporcionar servicios de seguridad. Para obtener más información sobre cómo escribir un programa de salida de canal, consulte Escritura de programas de salida de canal.

Seguridad a nivel de enlace mediante una salida de seguridad

Las salidas de seguridad suelen funcionar en pares: una en cada extremo de un canal. Se les llama inmediatamente después de que la negociación inicial de datos se ha completado en el inicio del canal.

Se pueden utilizar salidas de seguridad para proporcionar identificación y autenticación, control de accesos y confidencialidad.

Seguridad a nivel de enlace mediante una salida de mensajes

Una salida de mensajes sólo se puede utilizar en un canal de mensajes, no en un canal MQI. Tiene acceso tanto a la cabecera de colas de transmisión, MQXQH, que incluye el descriptor de mensaje incorporado, como a los datos de aplicación de un mensaje. Puede modificar el contenido del mensaje y cambiar su longitud.

Se puede utilizar una salida de mensajes para cualquier finalidad que requiera acceso al mensaje completo, más que a una parte del mismo.

Se pueden utilizar salidas de mensajes para proporcionar identificación y autenticación, control de accesos, confidencialidad, integridad de datos y servicio contra rechazos, y por motivos que no sean la seguridad.

Seguridad a nivel de enlace mediante salidas de emisión y recepción
Las salidas de emisión y recepción se pueden utilizar tanto en canales de mensajes como en canales MQI. Se les llama para todos los tipos de datos que fluyen en un canal y para flujos en ambas direcciones.

Las salidas de emisión y recepción tienen acceso a cada segmento de transmisión. Pueden modificar su contenido y cambiar su longitud.

En un canal de mensajes, si un MCA tiene que dividir un mensaje y enviarlo en más de un segmento de transmisión, se llama a una salida de emisión para cada segmento de transmisión que contiene una parte del mensaje y, en el extremo receptor, se llama a una salida de recepción para cada segmento de transmisión. Lo mismo sucede en un canal MQI si los parámetros de entrada o de salida de una llamada MQI son demasiado grandes como para que se envíen en un solo segmento de transmisión.

En un canal MQI, el byte 10 de un segmento de transmisión identifica la llamada MQI e indica si el segmento de transmisión contiene los parámetros de entrada o de salida de la llamada. Las salidas de emisión y recepción examinan este byte para determinar si la llamada MQI contiene datos de aplicación que se deban proteger.

Cuando se llama a una salida de emisión por primera vez, para adquirir e inicializar los recursos que necesita, puede solicitar al MCA que reserve una cantidad especificada de espacio en el almacenamiento intermedio que contiene un segmento de transmisión. Cuando se le llama posteriormente para procesar un segmento de transmisión, puede utilizar este espacio para añadir una clave cifrada o una firma digital, por ejemplo. La salida de recepción correspondiente en el otro extremo del canal puede eliminar los datos añadidos por la salida de emisión y utilizarlos para procesar el segmento de transmisión.

Las salidas de emisión y recepción son las más adecuadas en los casos en que no es necesario que comprendan la estructura de los datos que están manejando y, por lo tanto, pueden tratar cada segmento de transmisión como un objeto binario.

Se pueden utilizar salidas de emisión y recepción para proporcionar confidencialidad e integridad de datos, y por motivos que no sean la seguridad.

Tareas relacionadas
Identificación de la llamada API en un programa de salidas de envío o recepción

Seguridad a nivel de aplicación
La *seguridad a nivel de aplicación* hace referencia a los servicios de seguridad que se invocan en la interfaz entre una aplicación y un gestor de colas al que está conectada.

Estos servicios se invocan cuando la aplicación emite llamadas MQI dirigidas al gestor de colas. Los servicios los puede invocar, directa o indirectamente, la aplicación, el gestor de colas, otro producto que dé soporte a IBM MQ, o una combinación de cualquiera de esos productos que funcionen conjuntamente. La seguridad a nivel de aplicación se ilustra en la Figura 10 en la página 111.

La seguridad a nivel de aplicación se conoce también como *seguridad de extremo a extremo* o *seguridad a nivel de mensaje*.

Los siguientes son ejemplos de servicios de seguridad a nivel de aplicación:

- Cuando una aplicación transfiere un mensaje a una cola, el descriptor de mensaje contiene un ID de usuario asociado a la aplicación. No obstante, no hay datos presentes como, por ejemplo, una contraseña cifrada, que se puedan utilizar para autenticar el ID de usuario. Un servicio de seguridad puede añadir estos datos. Cuando la aplicación receptora recupera el mensaje, otro componente del servicio puede autenticar el ID de usuario utilizando los datos que ha transportado el mensaje. Este es un ejemplo de un servicio de identificación y autenticación.
- Un mensaje se puede cifrar cuando una aplicación lo transfiere a una cola y se puede descifrar cuando la aplicación receptora lo recupera.Este es un ejemplo de un servicio de confidencialidad.

- Un mensaje se puede comprobar cuando la aplicación receptora lo recupera. Esta comprobación determina si el contenido se ha modificado de forma deliberada ya que, en primer lugar, la aplicación emisora lo había transferido a la cola.Este es un ejemplo de un servicio de integridad de datos.

Planificación de Advanced Message Security
Advanced Message Security (AMS) es un componente de IBM MQ que proporciona un alto nivel de protección para los datos confidenciales que fluyen a través de la red IBM MQ, aunque no afecta a las aplicaciones finales.

Si está moviendo información delicada o valiosa, especialmente información confidencial o relacionada con los pagos como por ejemplo registros de pacientes o detalles de tarjetas de crédito, debe poner una atención especial en la seguridad de la información. Asegurarse de que la información que mueve la empresa conserva su integridad y está protegida frente al acceso no autorizado es un reto y una responsabilidad actual. También deberá cumplir con las regulaciones de seguridad, pudiendo sufrir sanciones en caso de no cumplirlas.

Puede desarrollar sus propias extensiones de seguridad para IBM MQ. Sin embargo, tales soluciones requieren habilidades especiales y pueden ser complicadas y caras de mantener. Advanced Message Security le ayuda a enfrentarse a estos retos al mover información dentro de la empresa entre virtualmente cualquier tipo de sistema de tecnologías de la información comercial.

Advanced Message Security amplía las características de seguridad de IBM MQ de las maneras siguientes:

- Proporciona protección de datos de principio a fin en el nivel de aplicación para su infraestructura de mensajes punto a punto, utilizando el cifrado o la firma digital de mensajes.
- Proporciona una seguridad exhaustiva sin escribir código de seguridad complejo ni modificar ni volver a compilar aplicaciones existentes.
- Utiliza la tecnología de Infraestructura de claves públicas (PKI) para proporcionar servicios de autenticación, autorización, confidencialidad e integridad de datos para mensajes.
- Proporciona la administración de políticas de seguridad para servidores distribuidos y de sistema principal.
- Soporta los servidores y los clientes de IBM MQ.
- Se integra con Managed File Transfer para proporcionar una solución de mensajería segura de principio a fin.

Para obtener más información, consulte “Advanced Message Security” en la página 610.

Cómo proporcionar su propia seguridad a nivel de aplicación
Puede proporcionar sus propios servicios de seguridad de nivel de aplicación. Para ayudarle a implementar la seguridad a nivel de aplicación, IBM MQ proporciona dos salidas, la salida de API y la salida cruzada de API.

La salida de API y la salida cruzada de API pueden proporcionar identificación y autenticación, control de accesos, confidencialidad, integridad de datos y servicios de no repudiación y otras funciones no relacionadas con la seguridad.

Si la salida de API o la salida cruzada de API no están soportadas en su entorno de sistema, es posible que desee considerar otros modos de proporcionar su propia seguridad a nivel de aplicación. Un método es desarrollar una API de nivel superior que encapsule la MQI. A continuación, los programadores utilizan esta API, en lugar de la MQI, para escribir aplicaciones IBM MQ.

Los motivos más comunes para utilizar una API de nivel superior son:

- Ocultar las funciones más avanzadas de la MQI a los programadores.
- Aplicar los estándares que utiliza la MQI.
- Añadir funciones a la MQI. Esta función adicional puede ser servicios de seguridad.

Algunos productos de proveedores utilizan esta técnica para proporcionar seguridad a nivel de aplicación para IBM MQ.

Si piensa proporcionar servicios de seguridad de este modo, tenga en cuenta lo siguiente en relación con la conversión de datos:

- Si se ha añadido una señal de seguridad, como por ejemplo una firma digital, a los datos de la aplicación contenidos en un mensaje, cualquier código que efectúe la conversión de datos deberá tener en cuenta la existencia de esta señal.
- Una señal de seguridad puede haberse derivado de una imagen binaria de los datos de aplicación. Por lo tanto, cualquier comprobación de la señal se debe realizar antes de convertir los datos.
- Si los datos de aplicación que contiene un mensaje se han cifrado, se deben descifrar antes de la conversión de datos.

Programas de salida de canal

Los *programas de salida de canal* son programas a los que se llama en lugares definidos de la secuencia de proceso de un MCA. Los usuarios y proveedores pueden escribir sus propios programas de salida de canal. IBM proporciona algunos de ellos.

Hay varios tipos de programas de salida de canal, pero sólo cuatro ofrecen seguridad a nivel de enlace:

- Salida de seguridad
- Salida de mensajes
- Salida de emisión
- Salida de recepción

Estos cuatro tipos de programa de salida de canal se ilustran en la Figura 11 en la página 116 y se describen en los temas siguientes.

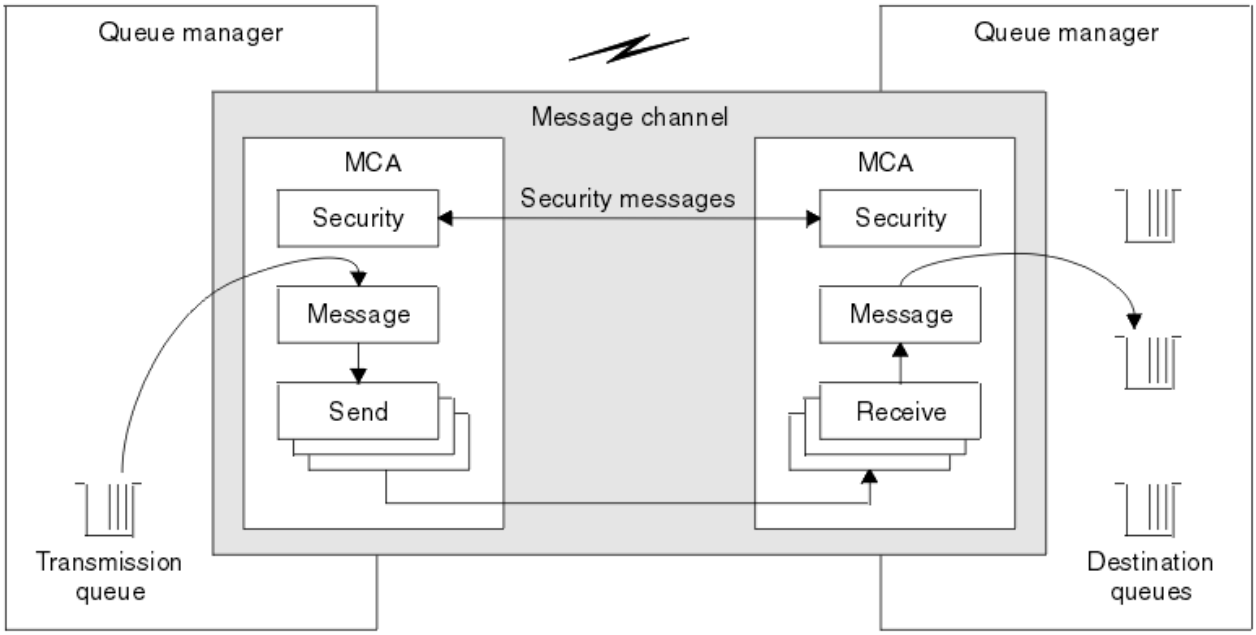


Figura 11. Salidas de seguridad, mensajes, emisión y recepción en un canal de mensajes

Conceptos relacionados

Programas de salida de canal para canales de mensajes

Visión general de las salidas de seguridad

Las salidas de seguridad normalmente funcionan en pares. Se les llama antes de que se inicie el flujo de mensajes y su finalidad es permitir que un MCA autentique su asociado.

Las *salidas de seguridad* suelen funcionar en pares; una en cada extremo de un canal. Se les llama inmediatamente después de que la negociación inicial de datos se ha completado en el inicio del canal,

pero antes de que los mensajes empiecen a fluir. El principal objetivo de la salida de seguridad es permitir que el MCA de cada extremo de un canal autentique su asociado. Sin embargo, no existe ningún método para evitar que una salida de seguridad lleve a cabo otra función, incluso funciones que no tienen nada que ver con la seguridad.

Las salidas de seguridad se pueden comunicar entre sí enviando *mensajes de seguridad*. El formato de un mensaje de seguridad no está definido y lo determina el usuario. Un posible resultado del intercambio de mensajes de seguridad es que una de las salidas de seguridad decida no continuar. En este caso, el canal se cierra y los mensajes no fluyen. Si hay una salida de seguridad en un solo extremo de un canal, se sigue llamando a la salida y esta puede decidir entre continuar o cerrar el canal.

Se puede llamar a las salidas de seguridad en canales de mensajes y de MQI. El nombre de una salida de seguridad se especifica como un parámetro en la definición de canal en cada extremo del canal.

Para obtener más información acerca de las salidas de seguridad, consulte la publicación "Seguridad a nivel de enlace mediante una salida de seguridad" en la página 113.

Salida de mensajes

Las salidas de mensajes solamente funcionan en canales de mensajes y normalmente funcionan en pares. Una salida de mensajes puede funcionar en todo el mensaje y realizar diversos cambios en el mismo.

Las *salidas de mensajes* en los extremos emisor y receptor de un canal suelen funcionar en pares. Se llama a una salida de mensajes en el extremo emisor de un canal después de que el MCA haya obtenido el mensaje de la cola de transmisión. En el extremo receptor de un canal, se llama a una salida de mensajes antes de que el MCA coloque un mensaje en su cola de destino.

Una salida de mensajes tiene acceso tanto a la cabecera de colas de transmisión, MQXQH, que incluye el descriptor de mensaje incorporado, como a los datos de aplicación en un mensaje. Una salida de mensajes puede modificar el contenido del mensaje y cambiar su longitud. Un cambio en la longitud puede dar lugar a la compresión, descompresión, cifrado o descifrado del mensaje. También puede dar lugar a la adición de datos al mensaje o a la eliminación de datos del mismo.

Las salidas de mensajes se pueden utilizar para cualquier objetivo que requiera acceso al mensaje completo, no a una parte del mismo, y no necesariamente por motivos de seguridad.

Una salida de mensajes puede determinar que el mensaje que está procesando actualmente no debe continuar hacia su destino. Luego el MCA transfiere el mensaje a la cola de mensajes no entregados. Una salida de mensajes también puede cerrar el canal.

Sólo se puede llamar a salidas de mensajes en canales de mensajes, no en canales MQI. Esto se debe a que el objetivo de un canal MQI es permitir que los parámetros de entrada y salida de las llamadas MQI fluyan entre la aplicación IBM MQ MQI client y el gestor de colas.

El nombre de una salida de mensajes se especifica como un parámetro de la definición de canal en cada extremo del canal. También puede especificar una lista de salidas de mensajes para que se ejecuten en sucesión.

Para obtener más información acerca de las salidas de mensajes, consulte la publicación "Seguridad a nivel de enlace mediante una salida de mensajes" en la página 113.

Salidas de emisión y recepción

Las salidas de emisión y recepción normalmente funcionan en pares. Actúan en segmentos de transmisión y es mejor utilizarlas cuando la estructura de los datos que están procesando no es relevante.

Una *salida de emisión* en un extremo de un canal y una *salida de recepción* en el otro extremo suelen funcionar en pares. Se llama a una salida de emisión justo antes de que un MCA emita un envío de comunicaciones para enviar datos a través de la conexión de comunicaciones. Se llama a una salida de recepción justo después de que un MCA haya vuelto a obtener el control que sigue a una recepción de comunicaciones y haya recibido datos de una conexión de comunicaciones. Si se utiliza la compartición de conversaciones, a través de un canal MQI, para cada conversación se llama a una instancia distinta de una salida de envío y recepción.

Los flujos del protocolo de canal de IBM MQ entre dos MCA en un canal de mensajes contienen información de control y datos del mensaje. De forma similar, en un canal MQI, los flujos contienen información de control, así como los parámetros de llamadas MQI. Se llama a salidas de emisión y recepción para todos los tipos de datos.

Los datos del mensaje fluyen en una sola dirección en un canal de mensajes pero, en un canal MQI, los parámetros de entrada de una llamada MQI fluyen en una dirección y los parámetros de salida fluyen en la otra. Tanto en los canales de mensajes como en los MQI, la información de control fluye en ambas direcciones. Como resultado, se puede llamar a salidas de emisión y de recepción en ambos extremos de un canal.

La unidad de datos que se transmite en un solo flujo entre dos MCA se denomina *segmento de transmisión*. Las salidas de emisión y recepción tienen acceso a cada segmento de transmisión. Pueden modificar su contenido y cambiar su longitud. Sin embargo, una salida de emisión no debe cambiar los 8 primeros bytes de un segmento de transmisión. Estos 8 bytes forman parte de la cabecera del protocolo de canal de IBM MQ. También hay restricciones en la cantidad en que una salida de emisión puede aumentar la longitud de un segmento de transmisión. En concreto, una salida de emisión no puede aumentar su longitud por encima del máximo negociado entre los dos MCA en el momento del inicio del canal.

En un canal de mensajes, si un mensaje es demasiado largo y no se puede enviar en un solo segmento de transmisión, el MCA emisor divide el mensaje y lo envía en más de un segmento de transmisión. Como consecuencia, se llama a una salida de emisión para cada segmento de transmisión que contiene una parte del mensaje y, en el extremo receptor, se llama a una rutina de recepción para cada segmento de transmisión. El MCA receptor vuelve a construir el mensaje a partir de los segmentos de transmisión después de que la salida de recepción los haya procesado.

De forma similar, en un canal MQI, los parámetros de entrada o salida de una llamada MQI se envían en más de un segmento de transmisión si son demasiado largos. Esto puede suceder, por ejemplo, en una llamada MQPUT, MQPUT1 o MQGET si los datos de aplicación son lo suficientemente grandes.

Teniendo esto en cuenta, es más adecuado utilizar salidas de emisión y recepción en casos en que no tengan que comprender la estructura de los datos que manejan y puedan, por tanto, tratar cada segmento de transmisión como un objeto binario.

Una salida de emisión o de recepción puede cerrar un canal.

Los nombres de una salida de emisión y de una de recepción se especifican como parámetros en la definición de canal en cada extremo de un canal. También puede especificar una lista de salidas de emisión para que se ejecuten en sucesión. De forma similar, puede especificar una lista de salidas de recepción.

Para obtener más información acerca de las salidas de recepción y de emisión, consulte la publicación “Seguridad a nivel de enlace mediante salidas de emisión y recepción” en la página 114.

Planificación de la integridad de datos

Planifique cómo preservar la integridad de los datos.

Puede implementar la integridad de datos a nivel de la aplicación o a nivel del enlace.

A nivel de aplicación, también puede utilizar los programas de salida de API si los recursos estándares no satisfacen los requisitos. Puede optar por utilizar Advanced Message Security (AMS) para firmar digitalmente los mensajes para protegerse con el fin de protegerlos frente la modificación no autorizada.

A nivel de enlace, puede optar por utilizar TLS, en cuyo caso debe planificar el uso de certificados digitales. También puede utilizar programas de salida de canal si los recursos estándares no satisfacen los requisitos.

Conceptos relacionados

“Protección de canales con SSL/TLS” en la página 122
El soporte de TLS en IBM MQ utiliza el objeto de información de autenticación del gestor de colas y diversos mandatos MQSC. También debe contemplar el uso de certificados digitales.

“Integridad de datos” en la página 10
El servicio de *integridad de datos* detecta si se han modificado los datos de forma no autorizada.

“Planificación de Advanced Message Security” en la página 115
Advanced Message Security (AMS) es un componente de IBM MQ que proporciona un alto nivel de protección para los datos confidenciales que fluyen a través de la red IBM MQ, aunque no afecta a las aplicaciones finales.

Referencia relacionada

Referencia a la salida de la API

Llamadas de salida de canal y estructuras de datos

Planificación de la auditoría

Decida qué datos necesita auditar, y cómo va a capturar y procesar información de auditoría. Tenga en cuenta cómo comprobar que el sistema está configurado correctamente.

Hay varios aspectos para la supervisión de la actividad. Los aspectos que debe tener en cuenta a menudo los definen los requisitos del auditor, y estas necesidades a menudo se controlan por los estándares normativos como HIPAA (Health Insurance Portability and Accountability Act) o SOX (Sarbanes-Oxley). IBM MQ proporciona características destinadas a ayudar con la conformidad con los estándares.

Considere si sólo está interesado en las excepciones o si está interesado en todo el comportamiento del sistema.

Algunos aspectos de la auditoría también pueden considerarse como la supervisión operativa; una distinción para la auditoría es que a menudo están examinando los datos históricos, no solo examinar las alertas en tiempo real. La supervisión se describe en la sección Supervisión y rendimiento.

¿Qué datos deben auditarse?

Considere qué tipos de datos o actividad es necesario auditar, tal como se describe en las secciones siguientes:

Los cambios realizados en IBM MQ utilizando las interfaces de IBM MQ
: Configure IBM MQ para emitir sucesos de instrumentación, específicamente sucesos de mandatos y sucesos de configuración.

Los cambios realizados en IBM MQ fuera de su control
: Algunos cambios pueden afectar a cómo se comporta IBM MQ, pero no pueden supervisarse directamente mediante IBM MQ. Algunos ejemplos de esos cambios incluyen cambios en la configuración de los archivos `mqs.ini`, `qm.ini`y `mqclient.ini`, la creación y supresión de gestores de colas, la instalación de los archivos binarios como programas de salida de usuario, y los cambios en los permisos de archivos. Para supervisar estas actividades, debe utilizar herramientas que se ejecutan en el nivel del sistema operativo. Hay diferentes herramientas disponibles y apropiadas para sistemas operativos diferentes. También puede tener registros creados por las herramientas asociadas como *sudo*.

El control operacional de IBM MQ
: Puede utilizar las herramientas del sistema operativo para auditar actividades como el inicio y la detención de gestores de colas. En algunos casos, IBM MQ se puede configurar para emitir sucesos de instrumentación.

La actividad de aplicación dentro de IBM MQ
: Para auditar las acciones de aplicaciones, por ejemplo la apertura de colas y la transferencia y obtención de mensajes, configure IBM MQ para emitir los sucesos adecuados.

Alertas de intrusos
: Para auditar las vulneraciones de la seguridad que se han intentado, configure el sistema para emitir sucesos de autorización. Los sucesos de canal también podrían ser útiles para mostrar actividad, especialmente si un canal finaliza inesperadamente.

Planificación de la captura, la visualización y el archivado de datos de auditoría

Muchos de los elementos necesarios se notifican como mensajes de sucesos de IBM MQ. Debe elegir herramientas que puedan leer y formatear estos mensajes. Si está interesado en el almacenamiento y análisis a largo plazo debe trasladarlos a un mecanismo de almacenamiento auxiliar como una base de datos. Si no procesa estos mensajes, estos permanecen en la cola de sucesos, posiblemente llenando la cola. Puede decidir implementar una herramienta que actúe automáticamente basándose en algunos sucesos; por ejemplo, emitir una alerta cuando se produce un fallo de seguridad.

Verificación de que el sistema está configurado correctamente

Se facilitan un conjunto de pruebas con IBM MQ Explorer. Utilícelas para comprobar si hay problemas en las definiciones de objetos.

Asimismo, compruebe periódicamente que la configuración del sistema es la que espera. Aunque los sucesos de mandatos y configuración pueden notificar cuando algo se modifica, también es útil para volcar la configuración y compararla con un buena copia conocida.

Planificación de seguridad según topología

En esta sección se describe la seguridad en situaciones específicas, en concreto de los canales, los clústeres de gestores de colas, las aplicaciones de publicación/suscripción y multidifusión, y cuando se utiliza un cortafuegos.

Consulte los subtemas siguientes para obtener más información:

Autorización de canal

Al enviar o recibir un mensaje a través de un canal, es necesario proporcionar acceso a diversos recursos de IBM MQ. Los agentes de canal de mensajes (MCA) son fundamentalmente aplicaciones IBM MQ que mueven mensajes entre gestores de colas y como tales requieren acceder a diversos recursos de IBM MQ para funcionar correctamente.

Para recibir mensajes en la hora de transferencia para los MCA, puede utilizar el ID de usuario asociado al MCA, o el ID de usuario asociado al mensaje.

En la hora de conexión puede correlacionar el ID de usuario certificado con un usuario alternativo, utilizando los registros de autenticación de canal **CHLAUTH**.

En IBM MQ, los canales pueden protegerse mediante el soporte TLS.

Los ID de usuario asociados con los canales emisores y receptores, excluido el canal emisor donde no se utiliza el atributo MCAUSER, requieren acceder a los siguientes recursos:

- El ID de usuario asociado a un canal emisor requiere acceso al gestor de colas, la cola de transmisión, la cola de mensajes no entregados y el acceso a los demás recursos requeridos por las salidas de canal.
- El ID de usuario MCAUSER de un canal receptor necesita la autorización *+setall*. La razón es que el canal receptor tiene que crear el MQMD completo, incluidos los campos de contexto, utilizando los datos que ha recibido del canal emisor remoto. Por lo tanto el gestor de colas requiere que el usuario que lleve a cabo esta actividad tenga la autorización *+setall*. Esta autorización *+setall* debe otorgarse al usuario para:
  - Todas las colas en las que el canal receptor coloca válidamente los mensajes.
  - El objeto de gestor de colas. Para obtener más información, consulte Autorizaciones para contexto.
- El ID de usuario MCAUSER de un canal receptor donde el originador ha solicitado un mensaje de informe COA necesita autorización *+passid* en la cola de transmisión que devuelve el mensaje de informe. Sin esta autorización, se anotan mensajes de error AMQ8077.
- Con el ID de usuario asociado al canal receptor, se pueden abrir las colas de destino para poner mensajes en ellas. Esto implica el uso de la MQI (interfaz de cola de mensajes), por lo tanto, es posible que sea necesario realizar comprobaciones de control de acceso adicionales si no está utilizando el Gestor de autorizaciones sobre objetos (OAM) de IBM MQ. Puede especificar si las comprobaciones de

autorización se realizan en el ID de usuario asociado al MCA (tal como se describe en este tema) o en el ID de usuario asociado al mensaje desde el campo UserIdentifier de MQMD).

Para los tipos de canal a los que se aplica, el parámetro **PUTAUT** de la definición de un canal especifica qué ID de usuario se utiliza para estas comprobaciones.

- De forma predeterminada, el canal utiliza la cuenta de servicio del gestor de colas, que tiene plenos derechos administrativos y no requiere autorizaciones especiales.
- En el caso e canales de conexión de servidor, las conexiones administrativas se bloquean de forma predeterminada por las reglas CHLAUTH y requieren un suministro explícito.
- Los canales de tipo receptor, peticionario y receptor en clúster permiten que cualquier gestor de colas adyacente realice la administración local, a menos el administrador tome medidas para restringir este acceso.
- No es necesario otorgar autoridad *dsp* y *ctrlx* al ID de usuario MCAUSER de un canal receptor.
- Antes de IBM MQ 8.0.0 Fix Pack 4, si se utiliza un ID de usuario al que le faltan privilegios administrativos de IBM MQ, hay que otorgar la autorización **dsp** y **ctrlx** a dicho ID de usuario para que el canal funcione.

  Desde IBM MQ 8.0.0 Fix Pack 4, no existe ninguna comprobación de autoridad cuando un canal se resincroniza a sí mismo y corrige números de secuencia.

  Sin embargo, la emisión de un mandato RESET CHANNEL sigue necesitando **+dsp** y **+ctrlx** en todos los releases.

  Atención: Cuando se requiere un restablecimiento de canal para la confirmación por lotes de mensajes, IBM MQ intenta consultar el canal, que requiere autorización **+dsp**.
- El atributo MCAUSER no se utiliza para el tipo de canal SDR.
- Si utiliza el ID de usuario asociado al mensaje, es probable que el ID de usuario proceda de un sistema remoto. Este ID de usuario del sistema remoto debe ser reconocido por el sistema de destino. Los mandatos siguientes son ejemplos del tipo de mandato que se puede ejecutar para otorgar autoridad a un ID de usuario de un sistema remoto:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect +inq +setall
```

```
setmqaut -m QMgrName -t chl -n Profile -g GroupName +dsp +ctrlx
```

donde *Perfil* es un canal.

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

donde *Perfil* es una cola de mensajes no entregados, si está configurada.

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

donde *Perfil* es una lista de colas autorizadas.

Atención: Tenga cuidado al autorizar un ID de usuario para que coloque mensajes en la cola de mandatos u otras colas del sistema sensibles.

El ID de usuario asociado al MCA depende del tipo del MCA. Hay dos tipos de MCA:

MCA de llamada
Los MCA que inician un canal. Los MCA de llamada se pueden iniciar como procesos individuales, como hebras del iniciador de canal o como hebras de una agrupación de procesos. El ID de usuario utilizado es el ID de usuario asociado al proceso padre (el iniciador de canal) o el ID de usuario asociado con el proceso que inicia al MCA.

MCA de respuesta
Los MCA de respuesta son los MCA que se inician como resultado de una solicitud del MCA de llamada. Los MCA de respuesta se pueden iniciar como procesos individuales, como hebras del

escucha o como hebras de una agrupación de procesos. El ID de usuario puede ser cualquiera de los tipos siguientes (en este orden de preferencia):

1. En APPC, el MCA de llamada puede indicar el ID de usuario que se debe utilizar para el MCA de respuesta. Esto se denomina el ID de usuario de red y se aplica solamente a los canales que se inician como procesos individuales. Establezca el ID de usuario de red ce con el parámetro USERID de la definición de canal.
2. Si no se utiliza el parámetro **USERID**, la definición de canal del MCA de respuesta puede especificar el ID de usuario que debe utilizar el MCA. Establezca el ID de usuario mediante el parámetro **MCAUSER** de la definición de canal.
3. Si el ID de usuario no se ha establecido siguiendo ninguno de los (dos) métodos anteriores, se utiliza el ID de usuario del proceso que inicia MCA o el ID de usuario del proceso padre (el escucha).

Conceptos relacionados

"Registros de autenticación de canal" en la página 53
Para ejercer un control más preciso sobre el acceso otorgado a la conexión de sistemas a nivel de canal, puede utilizar registros de autenticación de canal.

Referencia relacionada

Propiedades del registro de autenticación de canal

Protección de las definiciones de iniciador de canal

Sólo los miembros del grupo mqm pueden manipular iniciadores de canal.

Los iniciadores de canal de IBM MQ no son objetos IBM MQ; el OAM no controla el acceso a los mismos. IBM MQ no permite que los usuarios ni las aplicaciones manipulen estos objetos a menos que su ID de usuario sea miembro del grupo mqm. Si tiene una aplicación que emite el mandato PCF **StartChannelInitiator**, el ID de usuario especificado en el descriptor de mensaje del mensaje PCF debe ser miembro del grupo mqm en el gestor de colas de destino.

Un ID de usuario también debe ser miembro del grupo mqm en la máquina de destino para emitir los mandatos MQSC equivalentes mediante el mandato PCF de Escape o utilizando `runmqsc` en modalidad indirecta.

Colas de transmisión

Los gestores de colas transfieren automáticamente los mensajes remotos a una cola de transmisión; para ello no se requiere ninguna autorización especial.

Sin embargo, si tiene que transferir un mensaje directamente a una cola de transmisión, se necesita una autorización especial; consulte Tabla 12 en la página 140.

Salidas de canal

Si los registros de autenticación de canal no resultan adecuados, puede utilizar las salidas de canal para obtener una mayor seguridad. Una salida de seguridad forma una conexión segura entre dos programas de salida de seguridad. Un programa es para el agente de canal de mensajes (MCA) emisor y el otro es para el MCA receptor.

Consulte "Programas de salida de canal" en la página 116 para obtener más información sobre las salidas de canal.

Protección de canales con SSL/TLS

El soporte de TLS en IBM MQ utiliza el objeto de información de autenticación del gestor de colas y diversos mandatos MQSC. También debe contemplar el uso de certificados digitales.

Certificados digitales y depósitos de claves

Se recomienda establecer el atributo de etiqueta de certificado del gestor de colas (**CERTLABL**) en el nombre del certificado personal que se va a utilizar para la mayor parte de los canales, y alterarlo

temporalmente en el caso de excepciones, estableciendo la etiqueta de certificado en aquellos canales que requieren certificados diferentes.

Si necesita muchos canales con certificados que difieren del conjunto de certificados predeterminado en el gestor de colas, debe considerar la posibilidad de dividir los canales entre varios gestores de colas o utilizar un proxy MQIPT frente al gestor de colas para presentar un certificado diferente.

Puede utilizar un certificado diferente para cada canal, pero si almacena demasiados certificados en un depósito de claves, el rendimiento puede verse afectado cuando se inician los canales TLS. Intente mantener el número de certificados en un repositorio de claves en menos de unos 50 y considere que 100 es un máximo, ya que el rendimiento de IBM Global Security Kit (GSKit) disminuye drásticamente con repositorios de claves más grandes.

Si se permiten varios certificados en el mismo gestor de colas se aumentan las posibilidades de que se utilizarán varios certificados de CA en el mismo gestor de colas. De este modo, se aumentan las posibilidades de que existan conflictos en los espacios de nombres distinguidos del asunto para los certificados emitidos por diferentes entidades emisoras de certificados.

Aunque es probable que las entidades emisoras de certificados profesionales tengan mucho cuidado, normalmente las entidades emisoras de certificados internas no tienen convenios de denominación claros y pueden producirse incoherencias no intencionadas entre una entidad emisora de certificados y otra.

Debe comprobar el nombre distinguido del emisor del certificado además del nombre distinguido del asunto. Para ello,utilice un registro SSLPEERMAP de autenticación y establezca los campos **SSLPEER** y **SSLCERTI** de modo que coincidan con el nombre distinguido del asunto y del emisor respectivamente.

Certificados autofirmados y firmados por CA

Es importante planificar el uso de certificados digitales, tanto cuando se está desarrollando y probando la aplicación y para su uso en producción. Puede usar certificados firmados por CA o certificados autofirmados, en función del uso de los gestores de colas y las aplicaciones cliente.

Certificados firmados por CA
Para los sistemas de producción, obtenga los certificados de una autoridad certificadora de confianza (CA). Cuando obtiene un certificado de una CA externa, debe pagar por el servicio.

Certificados autofirmados
Mientras desarrolla la aplicación puede utilizar certificados autofirmados o certificados emitidos por una CA local, según la plataforma:

En sistemas AIX, Linux, and Windows , puede utilizar certificados autofirmados. Consulte “Creación de un certificado personal autofirmado en AIX, Linux, and Windows” en la página 551 para obtener instrucciones.

En los sistemas IBM i, puede utilizar los certificados firmados por la CA local. Consulte “Solicitar un certificado de servidor en IBM i” en la página 291 para obtener instrucciones.

En z/OS, puede utilizar certificados autofirmados o firmados por CA local. Consulte “Creating a self-signed personal certificate on z/OS” en la página 317 o “Requesting a personal certificate on z/OS” en la página 318 para obtener instrucciones.

Los certificados autofirmados no son adecuados para el uso en producción por las siguientes razones:

- Los certificados autofirmados no se pueden revocar, lo que podría permitir a un atacante suplantar una identidad después de que se haya comprometido una clave privada. Las CA pueden revocar un certificado comprometido, lo que impide su posterior uso. Los certificados firmados por una CA son, por lo tanto, más seguros para su uso en un entorno de producción, aunque los certificados autofirmados son más convenientes para un sistema de prueba.
- Los certificados autofirmados nunca caducan. Esto es cómodo y seguro en un entorno de prueba, pero en un entorno de producción pueden producirse infracciones de seguridad. El riesgo se agrava por el hecho de que los certificados autofirmados no se pueden revocar.

- Un certificado autofirmado se utiliza como un certificado personal y como un certificado de CA raíz (o ancla de confianza del certificado). Un usuario con un certificado personal autofirmado podría utilizarlo para firmar otros certificados personales. En general, esto no se cumple en certificados personales emitidos por una CA y representa un riesgo significativo.

CipherSpecs y certificados digitales

Únicamente un subconjunto de las CipherSpecs soportadas puede utilizarse con todos los tipos de certificados digitales soportados. Por consiguiente, es necesario que elija una CipherSpec adecuada para su certificado digital. Del mismo modo, si la política de seguridad de la empresa requiere que se utilice una determinada CipherSpec, debe obtener certificados digitales adecuados.

Para obtener más información sobre la relación entre CipherSpecs y los certificados digitales, consulte "Certificados digitales y compatibilidad de CipherSpec en IBM MQ" en la página 49

Políticas de validación de certificados

El estándar IETF RFC 5280 especifica una serie de reglas de validación de certificados que el software de aplicación compatible debe implementar para evitar ataques de suplantación. Un conjunto de reglas de validación de certificados se conoce como una política de validación de certificados. Para obtener más información sobre las políticas de validación de certificados en IBM MQ, consulte "Políticas de validación de certificados en IBM MQ" en la página 47.

Planificación de la comprobación de la revocación de certificados

Si se permiten varios certificados de diferentes entidades emisoras de certificados es muy probable que se requiera una comprobación adicional de la revocación de certificados.

En concreto, si ha configurado explícitamente el uso de un servidor de revocación desde una CA específica, por ejemplo, utilizando un objeto AUTHINFO o una estructura MQAIR (Authentication Information Record), la comprobación de la revocación fallará cuando se presente un certificado de una CA diferente.

Debe evitar esta configuración explícita del servidor de revocación de certificados. En su lugar, debe habilitar la comprobación implícita en la que cada certificado contiene su propia ubicación del servidor de revocación en una extensión del certificado, por ejemplo, un punto de distribución de CRL o AuthorityInfoAccess de OCSP.

Para obtener más información, consulte las secciones OCSPCheckExtensions y CDPCheckExtensions.

Mandatos y atributos para soporte TLS

El protocolo TLS (seguridad de la capa de transporte) proporciona seguridad de canal, con protección contra escuchas y manipulaciones no autorizadas y contra falsas identidades. El soporte de IBM MQ para TLS le permite especificar, en la definición de canal, que un canal determinado utilice seguridad TLS. También puede especificar información detallada sobre el tipo de seguridad que desea, como por ejemplo, el algoritmo de cifrado que desea utilizar.

- Los mandatos MQSC siguientes dan soporte a TLS:

  ALTER AUTHINFO
  Modifica los atributos de un objeto de información de autenticación.

  DEFINE AUTHINFO
  Crea un objeto de información de autenticación.

  DELETE AUTHINFO
  Suprime un objeto de información de autenticación.

  DISPLAY AUTHINFO
  Visualiza los atributos de un objeto de información de autenticación específico.

- Los siguientes parámetros de gestor de colas dan soporte a TLS:

CERTLABL
Define una etiqueta de certificado personal que utilizar.

KEYRPWD
En sistemas AIX, Linux, and Windows , define la contraseña que IBM MQ utiliza para acceder al repositorio de claves. Este campo se cifra utilizando el sistema de protección por contraseña.

SSLCRLNL
El atributo SSLCRLNL especifica una lista de nombres de objetos de información de autenticación que se utilizan para proporcionar ubicaciones de revocación de certificados para permitir la comprobación de certificados TLS mejorada.

SSLCRYP
En sistemas AIX, Linux, and Windows , establece el atributo de gestor de colas **SSLCryptoHardware** . Este atributo es el nombre de la serie de parámetros que puede utilizar para configurar el hardware criptográfico que tiene en el sistema.

SSLEV
Determina si se envía un mensaje de suceso TLS cuando un canal que utiliza TLS no puede establecer una conexión TLS correctamente.

SSLFIPS
Especifica si sólo se deben utilizar algoritmos certificados por FIPS si el cifrado se lleva a cabo en IBM MQ, en lugar de en el hardware de cifrado. Si el hardware de cifrado está configurado, se utilizan los módulos de cifrado que proporciona el producto de hardware, que pueden estar certificados por FIPS en un nivel determinado. Depende del producto de hardware que se esté utilizando.

SSLKEYR
En sistemas AIX, Linux, and Windows, asocia un repositorio de claves a un gestor de colas. GSKit le permite utilizar la seguridad TLS en sistemas AIX, Linux, and Windows .

SSLRKEYC
El número de bytes que se deben enviar y recibir en una conversación TLS antes de volver a negociar la clave secreta. El número de bytes incluye la información de control que envía el MCA.

- Los siguientes parámetros de canal dan soporte a TLS:

CERTLABL
Define una etiqueta de certificado personal que utilizar.

SSLCAUTH
Define si IBM MQ requiere y valida un certificado del cliente TLS.

SSLCIPH
Especifica la fuerza del cifrado y su función (CipherSpec), por ejemplo, TLS_RSA_WITH_AES_128_CBC_SHA. CipherSpec debe coincidir en ambos extremos del canal.

SSLPEER
Especifica el nombre distinguido (identificador exclusivo) de los asociados permitidos.

En este apartado se describen los mandatos **setmqaut**, **dspmqaut**, **dmpmqaut**, **rcrmqobj**, **rcdmqimg** y **dspmqfls** para dar soporte al objeto de información de autenticación. También describe los mandatos que se pueden utilizar para gestionar claves y certificados en AIX, Linux, and Windows. Consulte los apartados siguientes:

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- “Gestión de claves y certificados en AIX, Linux, and Windows” en la página 550

Para obtener una visión general de la seguridad de canal utilizando TLS, consulte

- “Protocolos de seguridad TLS en IBM MQ” en la página 25

Para conocer detalles de los mandatos MQSC asociados a TLS, consulte

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

Para conocer detalles de los mandatos PCF asociados a TLS, consulte

- Cambiar, copiar y crear un objeto de información de autenticación
- Suprimir objeto de información de autenticación
- Consultar objeto de información de autenticación

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 salida de canal” on page 116 for more information about channel exits.

Related tasks

Writing channel exit programs on z/OS

Servicios de seguridad SNA LU 6.2

SNA LU 6.2 ofrece cifrado a nivel de sesión, autenticación a nivel de sesión y autenticación a nivel de conversación.

Nota: En esta colección de temas se presupone que tiene conocimientos básicos sobre la Arquitectura de redes de sistemas (SNA). La otra documentación a la que se hace referencia en esta sección contiene una breve introducción a los conceptos y terminología relevantes. Si necesita una introducción técnica más completa a SNA, consulte el manual *Systems Network Architecture Technical Overview*, GC30-3073.

SNA LU 6.2 proporciona tres servicios de seguridad:

- Cifrado a nivel de sesión
- Autenticación a nivel de sesión
- Autenticación a nivel de conversación

Para el cifrado a nivel de sesión y la autenticación a nivel de sesión, SNA utiliza el algoritmo *Estándar de cifrado de datos (DES, Data Encryption Standard)*. El algoritmo DES es un algoritmo de cifrado de bloques que utiliza una clave simétrica para cifrar y descifrar datos. Tanto el bloque como la clave tienen una longitud de 8 bytes.

Cifrado a nivel de sesión

El *cifrado a nivel de sesión* cifra y descifra datos de sesión mediante el algoritmo DES. Por lo tanto, se puede utilizar para proporcionar un servicio de confidencial de enlace en canales SNA LU 6.2.

Las unidades lógicas (LU) pueden ofrecer cifrado de datos obligatorio (o necesario), cifrado de datos selectivo o ningún cifrado de datos.

En una *sesión de cifrado obligatorio*, una LU cifra todas las unidades de solicitud de datos de salida y descifra todas las unidades de solicitud de datos de entrada.

En una *sesión de cifrado selectivo*, una LU cifra sólo las unidades de solicitud de datos especificadas por el programa de transacción (TP) emisor. La LU emisora señala que los datos están cifrados estableciendo un indicador en la cabecera de la solicitud. Comprobando este indicador, la LU receptora puede indicar qué unidades de solicitud hay que descifrar antes de pasarlas al TP receptor.

En una red SNA, los MCA de IBM MQ son programas de transacciones. Los MCA no solicitan cifrado para ninguno de los datos que envían. Por lo tanto, el cifrado de datos selectivo no constituye una opción; sólo el cifrado de datos obligatorio o ningún cifrado de datos son opciones posibles en una sesión.

Para obtener información sobre cómo implementar el cifrado de datos obligatorio, consulte la documentación correspondiente a su subsistema SNA. Consulte la misma documentación para obtener información sobre formas más fuertes de cifrado que pueden estar disponibles para su uso en la plataforma, como por ejemplo el cifrado Triple DES de 24 bytes en z/OS.

Para obtener información más general sobre el cifrado a nivel de sesión, consulte el manual *Systems Network Architecture LU 6.2 Reference: Peer Protocols* , SC31-6808.

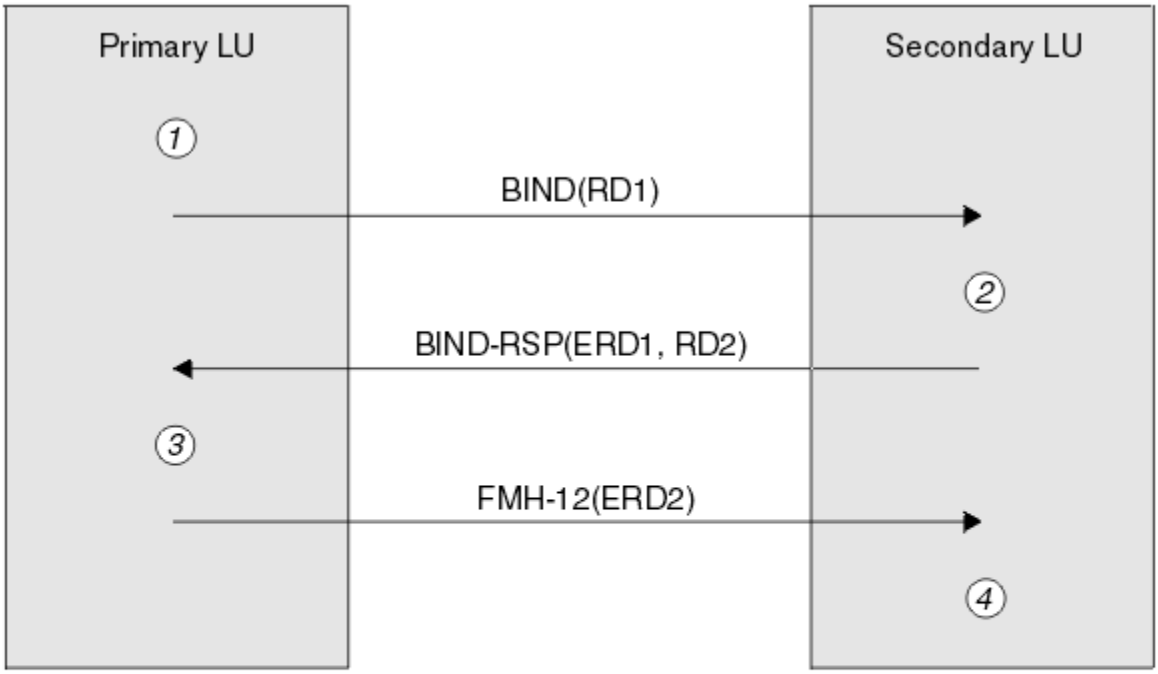
Autenticación a nivel de sesión

La *autenticación a nivel de sesión* es un protocolo de seguridad a nivel de sesión que permite que dos LU se identifiquen entre sí mientras están activando una sesión. También se denomina *verificación LU-LU*.

Puesto que una LU constituye realmente la "pasarela" a un sistema desde la red, es posible que considere que este nivel de autenticación es suficiente en determinadas circunstancias. Por ejemplo, si su gestor de colas tiene que intercambiar mensajes con un gestor de colas remoto que se está ejecutando en un entorno controlado y fiable, es posible que esté preparado para confiar en las identidades de los demás componentes del sistema remoto una vez autenticada la LU.

Cada LU consigue la autenticación a nivel de sesión verificando la contraseña de su asociado. La contraseña se denomina *contraseña LU-LU* porque se establece una contraseña entre cada par de LU. La forma en que se establece una contraseña LU-LU depende de la implementación y queda fuera del ámbito de SNA.

La Figura 12 en la página 129 ilustra los flujos correspondientes a la autenticación a nivel de sesión.



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. Flujos correspondientes a la autenticación a nivel de sesión

El protocolo correspondiente a la autenticación a nivel de sesión es el siguiente. Los números del procedimiento corresponden a los números de la Figura 12 en la página 129.

1. La LU principal genera un valor de datos aleatorios (RD1) y lo envía a la LU secundaria en la solicitud BIND.
2. Cuando la LU secundaria recibe la solicitud LU con los datos aleatorios, cifra los datos utilizando el algoritmo DES con su copia de la contraseña LU-LU como clave. Luego la LU secundaria genera un segundo valor de datos aleatorios (RD2) y lo envía, con los datos cifrados (ERD1), a la LU principal en la respuesta BIND.
3. Cuando la LU principal recibe la respuesta BIND, calcula su propia versión de los datos cifrados a partir de los datos aleatorios que ha generado originalmente. Para ello utiliza el algoritmo DES con su copia de la contraseña LU-LU como clave. Luego compara su versión con los datos cifrados recibidos en la respuesta BIND. Si los dos valores coinciden, la LU principal sabe que la LU secundaria tiene la misma contraseña que ella y la LU secundaria se autentica. Si los dos valores no coinciden, la LU principal finaliza la sesión.

   Luego la LU principal cifra los datos aleatorios que ha recibido en la respuesta BIND y envía los datos cifrados (ERD2) a la LU secundaria en una Cabecera de gestión de funciones 12 (FMH-12).
4. Cuando la LU secundaria recibe la FMH-12, calcula su propia versión de los datos cifrados a partir de los datos aleatorios que ha generado. Luego compara su versión con los datos cifrados que ha recibido en la FMH-12. Si los dos valores coinciden, la LU principal se autentica. Si los dos valores no coinciden, la LU secundaria finaliza la sesión.

En una versión mejorada del protocolo, que proporciona una mejor protección contra ataques de tipo "man in the middle" (hombre en medio), la LU secundaria calcula un Código de autenticación de mensaje

(MAC) de DES a partir de RD1, RD2 y el nombre completo de la LU secundaria, utilizando su copia de la contraseña LU-LU como clave. La LU secundaria envía el MAC a la LU principal en la respuesta BIND en lugar de ERD1.

La LU principal autentica la LU secundaria, calculando su propia versión del MAC, el cual compara con el MAC recibido en la respuesta BIND. Luego la LU principal calcula un segundo MAC a partir de RD1 y RD2 y envía el MAC a la LU secundaria en la FMH-12 en lugar de ERD2.

La LU secundaria autentica la LU principal calculando su propia versión del segundo MAC, el cual compara con el MAC recibido en la FMH-12.

Para obtener información sobre cómo configurar la autenticación a nivel de sesión, consulte la documentación de su subsistema SNA. Para obtener información más general sobre la autenticación a nivel de sesión, consulte el manual *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

Autenticación a nivel de conversación
Cuando un TP local intenta asignar una conversación con un TP asociado, la LU local envía una solicitud de adjuntar a la LU asociada, solicitándole que adjunte el TP asociado. Bajo determinadas circunstancias, la solicitud de adjuntar puede contener información de seguridad, la cual puede utilizar la LU asociada para autenticar el TP local. Esto se denomina *autenticación a nivel de conversación* o *verificación de usuario final*.

Los temas siguientes describen el modo en que IBM MQ proporciona soporte para la autenticación a nivel de conversación.

Para obtener más información acerca de la autenticación a nivel de conversación, consulte el manual *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

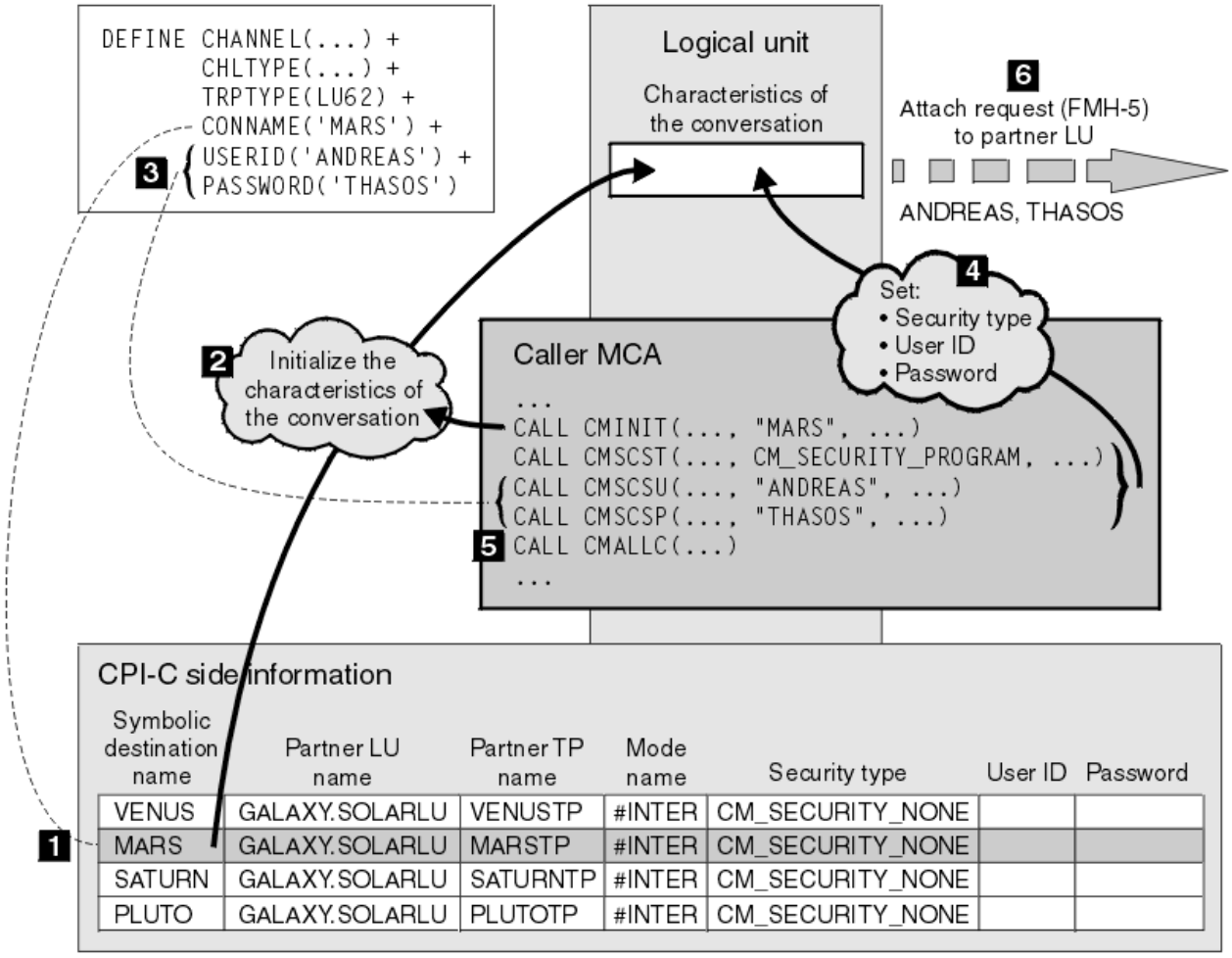
z/OS Para obtener información específica de z/OS, consulte z/OS MVS Planning: APPC/MVS Management.

Para obtener más información sobre CPI-C, consulte Utilización de comunicaciones CPI.

Para obtener más información sobre APPC/MVS TP Conversation Callable Services, consulte APPC/MVS TP Conversation Callable Services.

Multi *Soporte para la autenticación a nivel de conversación en Multiplatforms*
Utilice este tema para obtener una visión general de cómo funciona la autenticación a nivel de conversación en Multiplatforms.

El soporte para la autenticación a nivel de conversación en Multiplatforms se ilustra en Figura 13 en la página 131. Los números del diagrama corresponden a los números de la siguiente descripción.



Symbolic destination name	Partner LU name	Partner TP name	Mode name	Security type	User ID	Password
VENUS	GALAXY.SOLARLU	VENUSTP	#INTER	CM_SECURITY_NONE		
MARS	GALAXY.SOLARLU	MARSTP	#INTER	CM_SECURITY_NONE		
SATURN	GALAXY.SOLARLU	SATURNTP	#INTER	CM_SECURITY_NONE		
PLUTO	GALAXY.SOLARLU	PLUTOTP	#INTER	CM_SECURITY_NONE		

Figura 13. Soporte de IBM MQ para autenticación a nivel de conversación

En Multiplatforms, un MCA utiliza llamadas CPI-C (Common Programming Interface Communications) para comunicarse con un MCA asociado a través de una red SNA. En la definición de canal del extremo de llamada de un canal, el valor del parámetro CONNAME es un nombre de destino simbólico que identifica la entrada de información complementaria de CPI-C (1). Esta entrada especifica:

- El nombre de la LU asociada
- El nombre del TP asociado, que es un MCA de respuesta
- El nombre de la modalidad que se va a utilizar para la conversación

Una entrada de información complementaria también puede especificar la siguiente información de seguridad:

- Un tipo de seguridad.

  Los tipos de seguridad que se suelen implementar son CM_SECURITY_NONE, CM_SECURITY_PROGRAM y CM_SECURITY_SAME, pero hay otros definidos en la especificación CPI-C.
- Un ID de usuario.
- Una contraseña.

Un MCA de llamada se prepara para asignar una conversación con un MCA de respuesta, emitiendo la llamada CPI-C CMINIT, utilizando el valor de CONNAME como uno de los parámetros de la llamada. La llamada CMINIT identifica, como ayuda para la LU local, la entrada de información complementaria que el MCA tiene intención de utilizar para la conversación. La LU local utiliza los valores de esta entrada para inicializar las características de la conversación (2).

Luego el MCA de llamada comprueba los valores de los parámetros USERID y PASSWORD de la definición de canal (3). Si USERID está establecido, el MCA de llamada emite las siguientes llamadas CPI-C (4):

- CMSCST, para establecer el tipo de seguridad correspondiente a la conversación en CM_SECURITY_PROGRAM.
- CMSCSU, para establecer el ID de usuario correspondiente a la conversación en el valor de USERID.
- CMSCSP, para establecer la contraseña correspondiente a la conversación en el valor de PASSWORD. No se llama a CMSCSP a no ser que PASSWORD esté establecido.

El tipo de seguridad, ID de usuario y contraseña establecidos por estas llamadas prevalecen sobre los valores adquiridos previamente de la entrada de información complementaria.

Luego el MCA de llamada emite la llamada CPI-C CMALLC para asignar la conversación (5). En respuesta a esta llamada, la LU local envía una solicitud de adjuntar (Cabecera de gestión de funciones 5 o FMH-5) a la LU asociada (6).

Si la LU asociada acepta un ID de usuario y una contraseña, los valores de USERID y PASSWORD se incluyen en la solicitud de adjuntar. Si la LU asociada no acepta un ID de usuario y contraseña, los valores no se incluyen en la solicitud de adjuntar. La LU local descubre si la LU asociada va a aceptar un ID de usuario y contraseña como parte de un intercambio de información cuando las LU se vinculan para formar una sesión.

En una versión posterior de la solicitud de adjuntar, un sustituto de contraseña puede fluir entre las LU en lugar de una contraseña clara. Un sustituto de contraseña es un Código de autenticación de mensaje (MAC) de DES, o un resumen de mensaje SHA-1, formado a partir de la contraseña. Los sustitutos de contraseña sólo se pueden utilizar si ambas LU les dan soporte.

Cuando la LU asociada recibe una solicitud de adjuntar de entrada que contiene un ID de usuario y una contraseña, puede utilizar el ID de usuario y contraseña con finalidades de identificación y autenticación. Al hacer referencia a listas de control de accesos, la LU asociada también puede determinar si el ID de usuario tiene la autorización para asignar una conversación y adjuntar el MCA de respuesta.

Además, el MCA de respuesta se puede ejecutar bajo el ID de usuario que se incluye en la solicitud de adjuntar. En este caso, el ID de usuario se convierte en el ID de usuario predeterminado para el MCA de respuesta y se utiliza para comprobaciones de autorización cuando el MCA intenta conectar al gestor de colas. También se puede utilizar para siguientes comprobaciones de autorización cuando el MCA intenta acceder a los recursos del gestor de colas.

El modo en que se pueden utilizar un ID de usuario y una contraseña en la solicitud de adjuntar para identificación, autenticación y control de accesos depende de la implementación. Para obtener información específica de su subsistema SNA, consulte la documentación apropiada.

Si USERID no está establecido, el MCA de llamada no llama a CMSCST, CMSCSU ni CMSCSP. En este caso, la información de seguridad que fluye en una solicitud de adjuntar se determina únicamente por lo que se especifica en la entrada de información complementaria y por lo que la LU asociada aceptará.

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.

Seguridad para clústeres de gestores de colas

Aunque puede ser conveniente utilizar clústeres de gestores de colas, debe prestar especial atención a su seguridad.

Un *clúster de gestores de colas* es una red de gestores de colas asociados lógicamente de algún modo. Un gestor de colas que es miembro de un clúster se denomina un *gestor de colas de clúster*.

Una cola que pertenece a un gestor de colas de clúster se puede dar a conocer a otros gestores de colas del clúster. Dicha cola se denomina *cola de clúster*. Cualquier gestor de colas de un clúster puede enviar mensajes a colas de clúster sin necesidad de lo siguiente:

- Una definición de cola remota explícita para cada cola de clúster
- Canales definidos explícitamente a cada gestor de colas remoto y desde cada uno de ellos
- Una cola de transmisión individual para cada canal de salida

Puede crear un clúster en el que dos o varios gestores de colas sean clones. Esto significa que tienen instancias de las mismas colas locales, incluida cualquier cola local declarada como cola de clúster y que puede dar soporte a instancias de las mismas aplicaciones de servidor.

Cuando una aplicación conectada a un gestor de colas de clúster envía un mensaje a una cola de clúster que posee una instancia en cada uno de los gestores de colas clonados, IBM MQ decide a qué gestor de colas lo envía. Cuando muchas aplicaciones envían mensajes a una cola de clúster, IBM MQ equilibra la carga de trabajo entre todos los gestores de colas que poseen una instancia de la cola. Si uno de los sistemas que alberga un gestor de colas clonado sufre una anomalía, IBM MQ continúa equilibrando la carga de trabajo entre los gestores de colas restantes hasta que el sistema anómalo se reinicia.

Si va a utilizar clústeres de gestores de colas, debe tener en cuenta las siguientes cuestiones de seguridad:

- Si permite que solamente los gestores de colas seleccionados envíen mensajes al gestor de colas
- Si permite que solamente los usuarios seleccionados de un gestor de colas remoto envíen mensajes a una cola del gestor de colas
- Si permite que las aplicaciones conectadas al gestor de colas envíen mensajes solamente a las colas remotas seleccionadas

Estas consideraciones son relevantes incluso si no utiliza clústeres, pero resultan más importantes si los está utilizando.

Si una aplicación puede enviar mensajes a una cola de clúster, podrá enviar mensajes a cualquier otra cola de clúster sin necesitar definiciones de colas remotas, colas de transmisión ni canales adicionales. Por lo tanto, resulta más importante considerar si debe limitar el acceso a las colas de clúster en su gestor de colas y limitar las colas de clúster a aquéllas a las que sus aplicaciones pueden enviar mensajes.

Hay algunas consideraciones de seguridad adicionales que resultan relevantes solamente si está utilizando clústeres de gestores de colas:

- Si permite que solamente los gestores de colas seleccionados se unan a un clúster
- Forzar que los gestores de colas no deseados abandonen un clúster

Para obtener más información sobre todas estas consideraciones, consulte Mantenimiento de la seguridad de los clústeres. z/OS Si desea ver consideraciones específicas de IBM MQ for z/OS, consulte “Security in queue manager clusters on z/OS” en la página 268.

Tareas relacionadas

"Cómo impedir que los gestores de colas reciban mensajes" en la página 490
Puede impedir que un gestor de colas reciba mensajes si no está autorizado para recibirlos utilizando programas de salida.

Seguridad para Publicación/Suscripción de IBM MQ

Existen consideraciones de seguridad adicionales si está utilizando Publicación/Suscripción de IBM MQ.

En un sistema de publicación/suscripción, hay dos tipos de aplicaciones: el publicador y el suscriptor. Los *publicadores* proporcionan información en forma de mensajes IBM MQ. Cuando un publicador publica un mensaje, especifica un *tema*, que identifica el tema de la información que contiene el mensaje.

Los *suscriptores* son los que consumen la información publicada. Un suscriptor especifica los temas que le interesan suscribiéndose a ellos.

El *gestor de colas* es una aplicación que se suministra con Publicación/Suscripción de IBM MQ. Éste recibe los mensajes que han publicado los publicadores y las peticiones de suscripción de los suscriptores y dirige los mensajes publicados a los suscriptores. A un suscriptor se le envían solamente los mensajes de los temas a los que se ha suscrito.

Para obtener más información, consulte Seguridad de Publicación/suscripción.

Seguridad de multidifusión

Utilice esta información para comprender por qué pueden ser necesarios los procesos de seguridad con IBM MQ Multicast.

IBM MQ Multicast no tiene seguridad incorporada. Las comprobaciones de seguridad se manejan en el gestor de colas en tiempo de MQOPEN y el valor del campo MQMD lo maneja el cliente. Es posible que algunas aplicaciones de la red no sean aplicaciones de IBM MQ (por ejemplo, las aplicaciones LLM, consulte Interoperatividad de multidifusión con mensajes de baja latencia de IBM MQ para obtener más información), tal vez tenga que implementar sus propios procedimientos de seguridad porque las aplicaciones receptoras no pueden estar seguras de la validez de los campos de contexto.

Hay tres procesos de seguridad que se deben tener en cuenta:

Control de accesos

El control de acceso en IBM MQ está basado en los ID de usuario. Para obtener más información sobre este asunto, consulte "Control de accesos para clientes" en la página 108.

Seguridad de red

Una red aislada podría ser una opción de seguridad viable para evitar mensajes falsos. Es posible que una aplicación en la dirección del grupo de multidifusión publique mensajes malintencionados utilizando las funciones de comunicación nativas, que son imposibles de distinguir de los mensajes MQ porque vienen de una aplicación en la misma dirección del grupo de multidifusión.

También es posible que un cliente en la dirección del grupo de multidifusión reciba mensajes que estaban previstos para otros clientes en la misma dirección del grupo de multidifusión.

Aislar la red de multidifusión asegura que sólo los clientes y aplicaciones válidos tienen acceso. Esta precaución de seguridad puede impedir que entren mensajes malintencionados y que salga información confidencial.

Para obtener información sobre las direcciones de red de grupo de multidifusión, consulte: Establecer la red adecuada para el tráfico de multidifusión

Firmas digitales

Una firma digital se crea cifrando una representación de un mensaje. El cifrado utiliza la clave privada del que firma y, por motivos prácticos, suele operar en un resumen del mensaje en lugar de hacerlo en el mensaje propiamente dicho. La firma digital de un mensaje antes de MQPUT es una buena precaución de seguridad, pero este proceso puede tener un efecto perjudicial sobre el rendimiento si hay un gran volumen de mensajes.

Las firmas digitales varían con los datos que se firman. Si la misma entidad firma digitalmente dos mensajes diferentes, las dos firmas serán diferentes pero ambas pueden verificarse con la misma clave pública, es decir, la clave pública de la entidad que ha firmado los mensajes.

Como se ha mencionado anteriormente en esta sección, puede ser posible que una aplicación de la dirección del grupo de multidifusión publique mensajes malintencionados utilizando las funciones de comunicación nativas, que son imposibles de distinguir de los mensajes MQ. Las firmas digitales proporcionan una prueba de origen y solamente el emisor conoce la clave privada, que proporciona una prueba clara de que el remitente es el originador del mensaje.

Para obtener más información sobre este asunto, consulte "Conceptos de cifrado" en la página 11.

Cortafuegos y IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru puede simplificar la comunicación a través de un cortafuegos.

MQIPT permite que dos gestores de colas intercambien mensajes, o que una aplicación cliente IBM MQ se conecte a un gestor de colas, sin necesidad de una conexión TCP/IP directa. Esta arquitectura es útil si un cortafuegos prohíbe una conexión TCP/IP directa entre dos sistemas. El uso de MQIPT como proxy puede hacer que el paso de datos de canal de IBM MQ a través de un cortafuegos sea más sencillo y más gestionable. MQIPT también puede proteger los datos de IBM MQ que se envían a través de Internet utilizando TLS (Transport Layer Security) y los datos IBM MQ de túnel dentro de HTTP.

Para obtener más información, 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 194.

Refer to other sections for details, in particular "Profiles used to control access to IBM MQ resources" on page 204.

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 199.
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 260.

- **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 260.
   - **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 autenticación 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 seguridad TLS en 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 268

Configuración de seguridad

Esta colección de temas contiene información específica para distintos sistemas operativos y para el uso de clientes.

ALW Configuración de la seguridad en AIX, Linux, and Windows

Consideraciones de seguridad específicas a sistemas AIX, Linux, and Windows.

Los gestores de colas de IBM MQ transfieren información que puede ser muy valiosa, por lo que necesita utilizar un sistema de autorización para asegurar que los usuarios no autorizados no puedan acceder a sus gestores de colas. Contemple los siguientes tipos de controles de seguridad:

Quién puede administrar IBM MQ
Puede definir el conjunto de usuarios que puede emitir mandatos para administrar IBM MQ.

Quién puede utilizar objetos IBM MQ
Puede definir qué usuarios (generalmente aplicaciones) pueden utilizar llamadas MQI y mandatos PCF para realizar lo siguiente:

- Quién puede conectarse a un gestor de colas.
- Quién puede acceder a objetos (colas, definiciones de proceso, listas de nombres, canales, canales de conexión de cliente, escuchas, servicios, procesos y objetos de información de autenticación) y qué tipos de acceso tienen a dichos objetos.
- Quién puede acceder a mensajes de IBM MQ.
- Quién puede acceder a la información de contexto asociada a un mensaje.

Seguridad de canal
Debe asegurarse de que los canales que se utilizan para enviar mensajes a sistemas remotos puedan acceder a los recursos necesarios.

Puede utilizar los recursos operativos estándar para otorgar acceso a las bibliotecas de programa, las bibliotecas de enlaces de la MQI y a los mandatos. Sin embargo, el directorio que contiene las colas y otros datos de los gestores de colas es privado para IBM MQ; no utilice mandatos estándar del sistema operativo para otorgar o revocar autorizaciones a los recursos de la MQI.

ALW Cómo funcionan las autorizaciones en AIX, Linux, and Windows

Las tablas de especificación de autorizaciones de los temas de esta sección definen de forma precisa cómo funcionan las autorizaciones y las restricciones que se aplican.

Las tablas se aplican a estas situaciones:

- Aplicaciones que emiten llamadas MQI
- Programas de administración que emiten mandatos MQSC como mandatos PCF de escape
- Programas de administración que emiten mandatos PCF

En esta sección, la información se presenta como un conjunto de tablas que especifican lo siguiente:

Acción que se va a realizar
Opción MQI, mandato MQSC o mandato PCF.

Objeto de control de acceso
Cola, proceso, gestor de colas, lista de nombres, información de autenticación, canal, canal de conexión cliente, receptor o servicio.

Autorización necesaria
Expresada como constante de tipo MQZAO_.

En las tablas, las constantes que tienen el prefijo MQZAO_ corresponden a las palabras claves de la lista de autorizaciones del mandato `setmqaut` para la entidad específica. Por ejemplo, MQZAO_BROWSE corresponde a la palabra clave `+browse`, MQZAO_SET_ALL_CONTEXT corresponde a la palabra clave `+setall`, etc. Estas constantes están definidas en el archivo de cabecera cmqzc.h que se proporciona con el producto.

ALW *Autorizaciones para llamadas MQI*

MQCONN, **MQOPEN**, **MQPUT1** y **MQCLOSE** pueden requerir comprobaciones de autorización. Las tablas de este tema muestran un resumen de las autorizaciones necesarias para cada llamada.

Una aplicación puede emitir determinadas llamadas y opciones MQI sólo si el identificador de usuario bajo el que se está ejecutando (o cuyas autorizaciones puede asumir) tiene la autorización pertinente.

Hay cuatro llamadas MQI que pueden requerir comprobaciones de autorización: **MQCONN**, **MQOPEN**, **MQPUT1** y **MQCLOSE**.

Para **MQOPEN** y **MQPUT1**, la comprobación de autorización se efectúa en el nombre del objeto que se está abriendo, y no en el nombre o nombres resultantes de la resolución del nombre. Por ejemplo, una aplicación puede tener autorización para abrir una cola alias sin tener autorización para abrir la cola base en la que se resuelve la cola alias. La regla es que la comprobación se lleva a cabo en la primera definición encontrada durante el proceso de resolución de un nombre que no es un alias de gestor de colas, a menos que la definición de alias de gestor de colas se abra directamente; es decir, su nombre se visualiza en el campo *ObjectName* del descriptor de objeto. Para el objeto que se va a abrir siempre es necesario tener autorización. En algunos casos, también se necesita una autorización adicional independiente de la cola que se obtiene a través de una autorización para el objeto de gestor de colas.

Tabla 10 en la página 139, Tabla 11 en la página 139, Tabla 12 en la página 140 y Tabla 13 en la página 141 resumen las autorizaciones necesarias para cada llamada. La indicación *No es aplicable* significa que la comprobación de autorización no está asociada a esta operación. La indicación *No se comprueba* significa que no se realiza una comprobación de autorización.

Nota: En estas tablas no se mencionan listas de nombres, canales, canales de conexión de cliente, escuchas, servicios u objetos de información de autenticación. Esto se debe a que ninguna de las autorizaciones se aplica a estos objetos, salvo MQOO_INQUIRE, para la que se aplican las mismas autorizaciones que para los demás objetos.

La autorización especial MQZAO_ALL_MQI incluye todas las autorizaciones de las tablas que sean relevantes al tipo de objeto, excepto MQZAO_DELETE y MQZAO_DISPLAY, que están clasificadas como autorizaciones de administración.

Para poder modificar cualquier opción de contexto de mensaje, debe tener las autorizaciones apropiadas para emitir la llamada. Por ejemplo, para poder utilizar MQOO_SET_IDENTITY_CONTEXT o MQPMO_SET_IDENTITY_CONTEXT, debe tener el permiso `+setid`.

Tabla 10. Autorización de seguridad necesaria para llamadas MQCONN

Autorización necesaria para:	**Objeto de cola (“1” en la página 141)**	**Objeto de proceso**	**Objeto gestor de colas**
MQCONN	No aplicable	No aplicable	MQZAO_CONNECT

Tabla 11. Autorización de seguridad necesaria para llamadas MQOPEN

Autorización necesaria para:	**Objeto de cola (“1” en la página 141)**	**Objeto de proceso**	**Objeto gestor de colas**
MQOO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE
MQOO_BROWSE	MQZAO_BROWSE	No aplicable	No se comprueba
MQOO_INPUT_*	MQZAO_INPUT	No aplicable	No se comprueba
MQOO_SAVE_ALL_CONTEXT **(“2” en la página 141)**	MQZAO_INPUT	No aplicable	No aplicable
MQOO_OUTPUT (Cola normal) **(“3” en la página 141)**	MQZAO_OUTPUT	No aplicable	No aplicable
MQOO_PASS_IDENTITY_CONTEXT **(“4” en la página 141)**	MQZAO_PASS_IDENTITY_CONTEXT	No aplicable	No se comprueba

Tabla 11. Autorización de seguridad necesaria para llamadas MQOPEN (continuación)			
Autorización necesaria para:	**Objeto de cola (“1” en la página 141)**	**Objeto de proceso**	**Objeto gestor de colas**
MQOO_PASS_ALL_ CONTEXT **(“4” en la página 141, “5” en la página 141)**	MQZAO_PASS _ALL_CONTEXT	No aplicable	No se comprueba
MQOO_SET_ IDENTITY_CONTEXT **(“4” en la página 141, “5” en la página 141)**	MQZAO_SET_ IDENTITY_CONTEXT	No aplicable	MQZAO_SET_ IDENTITY_CONTEXT **(“6” en la página 141)**
MQOO_SET_ ALL_CONTEXT **(“4” en la página 141, “7” en la página 141)**	MQZAO_SET_ ALL_CONTEXT	No aplicable	MQZAO_SET_ ALL_CONTEXT **(“6” en la página 141)**
MQOO_OUTPUT (cola de transmisión) **(“8” en la página 141)**	MQZAO_SET_ ALL_CONTEXT	No aplicable	MQZAO_SET_ ALL_CONTEXT **(“6” en la página 141)**
MQOO_SET	MQZAO_SET	No aplicable	No se comprueba
MQOO_ALTERNATE_ USER_AUTHORITY	**(“9” en la página 141)**	**(“9” en la página 141)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(“9” en la página 141, “10” en la página 141)**

Tabla 12. Autorización de seguridad necesaria para llamadas MQPUT1			
Autorización necesaria para:	**Objeto de cola (“1” en la página 141)**	**Objeto de proceso**	**Objeto gestor de colas**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(“11” en la página 141)**	No aplicable	No se comprueba
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(“11” en la página 141)**	No aplicable	No se comprueba
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(“11” en la página 141)**	No aplicable	MQZAO_SET_ IDENTITY_CONTEXT **(“6” en la página 141)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(“11” en la página 141)**	No aplicable	MQZAO_SET_ ALL_CONTEXT **(“6” en la página 141)**
(Cola de transmisión) **(“8” en la página 141)**	MQZAO_SET_ ALL_CONTEXT	No aplicable	MQZAO_SET_ ALL_CONTEXT **(“6” en la página 141)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(“12” en la página 141)**	No aplicable	MQZAO_ALTERNATE_ USER_AUTHORITY **(“10” en la página 141)**

Tabla 13. Autorización de seguridad necesaria para llamadas MQCLOSE			
Autorización necesaria para:	**Objeto de cola (“1” en la página 141)**	**Objeto de proceso**	**Objeto gestor de colas**
MQCO_DELETE	MQZAO_DELETE **(“13” en la página 141)**	No aplicable	No aplicable
MQCO_DELETE _PURGE	MQZAO_DELETE **(“13” en la página 141)**	No aplicable	No aplicable

Notas para las tablas:

1. Si se está abriendo una cola modelo:
   - Para la cola modelo, es necesaria la autorización MQZAO_DISPLAY además de la autorización para abrir la cola modelo correspondiente al tipo de acceso para el que se está efectuando la apertura.
   - La autorización MQZAO_CREATE no es necesaria para crear la cola dinámica.
   - El identificador de usuario utilizado para abrir la cola modelo se otorga automáticamente a todas las autorizaciones específicas de la cola (equivalentes a MQZAO_ALL) para la cola dinámica creada.
2. También debe especificarse MQOO_INPUT_*. Esto es válido para una cola local, modelo o alias.
3. Esta comprobación se realiza en todas las salidas, excepto en las colas de transmisión (vea la nota “8” en la página 141).
4. También debe especificarse MQOO_OUTPUT.
5. Esta opción también implica MQOO_PASS_IDENTITY_CONTEXT.
6. Esta autorización es necesaria tanto para el objeto gestor de colas como para la cola concreta.
7. Esta opción también implica MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT y MQOO_SET_IDENTITY_CONTEXT.
8. Esta comprobación se realiza para una cola local o modelo cuyo atributo de cola *Usage* sea MQUS_TRANSMISSION y se esté abriendo directamente para salida. Esto no es aplicable si se abre una cola remota (especificando los nombres del gestor de colas remoto y la cola remota, o especificando el nombre de una definición local de la cola remota).
9. También debe especificarse como mínimo una de las opciones MQOO_INQUIRE (para cualquier tipo de objeto) o MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT o MQOO_SET (para las colas). La comprobación que se lleva a cabo es la misma que en las otras opciones especificadas, utilizando el identificador de usuario alternativo suministrado para la autorización sobre el objeto específico nombrado, y la autorización sobre la aplicación actual para la comprobación MQZAO_ALTERNATE_USER_IDENTIFIER.
10. Esta autorización permite especificar cualquier *AlternateUserId*.
11. También se realiza una comprobación MQZAO_OUTPUT si la cola no tiene un atributo de cola *Usage* de MQUS_TRANSMISSION.
12. La comprobación que se lleva a cabo es la misma que en la otras opciones especificadas, utilizando el identificador de usuario alternativo suministrado para la autorización sobre la cola específica nombrada, y la autorización sobre la aplicación actual para la comprobación MQZAO_ALTERNATE_USER_IDENTIFIER.
13. La comprobación solo se lleva a cabo si se cumplen las dos sentencias siguientes:
    - Se está cerrando y suprimiendo una cola dinámica permanente.
    - La cola no la ha creado la llamada MQOPEN que ha devuelto el descriptor de objeto usado.

    De lo contrario, no hay comprobación.

ALW *Autorizaciones para mandatos MQSC en los PCF de escape*

Esta información resume las autorizaciones necesarias para cada mandato MQSC contenido en un PCF de escape.

No es aplicable significa que esta operación no tiene sentido en este tipo de objeto.

El ID de usuario bajo el que se ejecuta el programa que envía el mandato también debe tener las autorizaciones siguientes:

- Autorización MQZAO_CONNECT para el gestor de colas
- Autorización MQZAO_DISPLAY sobre el gestor de colas para realizar mandatos PCF
- Autorización para emitir los mandatos MQSC en el texto del mandato PCF de Escape

ALTER *objeto*

Objeto	**Autorización necesaria**
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	MQZAO_CHANGE
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE
Información de comunicación	MQZAO_CHANGE

CLEAR *objeto*

Objeto	**Autorización necesaria**
Cola	MQZAO_CLEAR
Tema	MQZAO_CLEAR
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	No aplicable
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable
Información de comunicación	No aplicable

DEFINE *objeto* NOREPLACE (“1” en la página 146)

Objeto	**Autorización necesaria**
Cola	MQZAO_CREATE **(“2” en la página 146)**
Tema	MQZAO_CREATE **(“2” en la página 146)**

Objeto	Autorización necesaria
Proceso	MQZAO_CREATE **(“2” en la página 146)**
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CREATE **(“2” en la página 146)**
Información de autenticación	MQZAO_CREATE **(“2” en la página 146)**
Canal	MQZAO_CREATE **(“2” en la página 146)**
Canal de conexión de cliente	MQZAO_CREATE **(“2” en la página 146)**
Escucha	MQZAO_CREATE **(“2” en la página 146)**
Servicio	MQZAO_CREATE **(“2” en la página 146)**
Información de comunicación	MQZAO_CREATE **(“2” en la página 146)**

DEFINE *objeto* REPLACE (“1” en la página 146, “3” en la página 146)

Objeto	Autorización necesaria
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE
Información de comunicación	MQZAO_CHANGE

DELETE *objeto*

Objeto	Autorización necesaria
Cola	MQZAO_DELETE
Tema	MQZAO_DELETE
Proceso	MQZAO_DELETE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_DELETE
Información de autenticación	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de conexión de cliente	MQZAO_DELETE
Escucha	MQZAO_DELETE
Servicio	MQZAO_DELETE

Objeto	Autorización necesaria
Información de comunicación	MQZAO_DELETE

DISPLAY *objeto*

Objeto	Autorización necesaria
Cola	MQZAO_DISPLAY
Tema	MQZAO_DISPLAY
Proceso	MQZAO_DISPLAY
Gestor de colas	MQZAO_DISPLAY
Lista de nombres	MQZAO_DISPLAY
Información de autenticación	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de conexión de cliente	MQZAO_DISPLAY
Escucha	MQZAO_DISPLAY
Servicio	MQZAO_DISPLAY
Información de comunicación	MQZAO_DISPLAY

START *objeto*

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	MQZAO_CONTROL
Servicio	MQZAO_CONTROL
Información de comunicación	No aplicable

STOP *objeto*

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable

Objeto	Autorización necesaria
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	MQZAO_CONTROL
Servicio	MQZAO_CONTROL
Información de comunicación	No aplicable

Mandatos de canal

Mandato	Objeto	Autorización necesaria
PING CHANNEL	Canal	MQZAO_CONTROL
RESET CHANNEL	Canal	MQZAO_CONTROL_EXTENDED
RESOLVE CHANNEL	Canal	MQZAO_CONTROL_EXTENDED

Mandatos de suscripción

Mandato	Objeto	Autorización necesaria
ALTER SUB	Tema	MQZAO_CONTROL
DEFINE SUB	Tema	MQZAO_CONTROL
DELETE SUB	Tema	MQZAO_CONTROL
DISPLAY SUB	Tema	MQZAO_DISPLAY

Mandatos de seguridad

Mandato	Objeto	Autorización necesaria
SET AUTHREC	Gestor de colas	MQZAO_CHANGE
DELETE AUTHREC	Gestor de colas	MQZAO_CHANGE
DISPLAY AUTHREC	Gestor de colas	MQZAO_DISPLAY
DISPLAY AUTHSERV	Gestor de colas	MQZAO_DISPLAY
DISPLAY ENTAUTH	Gestor de colas	MQZAO_DISPLAY
SET CHLAUTH	Gestor de colas	MQZAO_CHANGE
DISPLAY CHLAUTH	Gestor de colas	MQZAO_DISPLAY
REFRESH SECURITY	Gestor de colas	MQZAO_CHANGE

Muestra el estado

Mandato	Objeto	Autorización necesaria
DISPLAY CHSTATUS	Gestor de colas	MQZAO_DISPLAY Tenga en cuenta que la autorización `+inq` (o equivalente MQZAO_INQUIRE) es necesaria en la cola de transmisión si el tipo de canal es CLUSSDR.
DISPLAY LSSTATUS	Gestor de colas	MQZAO_DISPLAY
DISPLAY PUBSUB	Gestor de colas	MQZAO_DISPLAY
DISPLAY SBSTATUS	Gestor de colas	MQZAO_DISPLAY
DISPLAY SVSTATUS	Gestor de colas	MQZAO_DISPLAY
DISPLAY TPSTATUS	Gestor de colas	MQZAO_DISPLAY

Mandatos de clúster

Mandato	Objeto	Autorización necesaria
DISPLAY CLUSQMGR	Gestor de colas	MQZAO_DISPLAY
REFRESH CLUSTER	grupo de pertenencia 'mqm' necesario	
RESET CLUSTER	grupo de pertenencia 'mqm' necesario	
SUSPEND QMGR	grupo de pertenencia 'mqm' necesario	
RESUME QMGR	grupo de pertenencia 'mqm' necesario	

Otros mandatos administrativos

Mandato	Objeto	Autorización necesaria
PING QMGR	Gestor de colas	MQZAO_DISPLAY
REFRESH QMGR	Gestor de colas	MQZAO_CHANGE
RESET QMGR	Gestor de colas	MQZAO_CHANGE
DISPLAY CONN	Gestor de colas	MQZAO_DISPLAY
STOP CONN	Gestor de colas	MQZAO_CHANGE

Nota:

1. Para los mandatos DEFINE, se necesita también la autorización MQZAO_DISPLAY sobre el objeto LIKE, si se ha especificado uno, o sobre el objeto SYSTEM.DEFAULT.xxx adecuado si se ha omitido LIKE.
2. La autorización MQZAO_CREATE no es específica de un objeto o tipo de objeto en particular. Para un gestor de colas especificado, la autorización de creación se otorga para todos los objetos, especificando un tipo de objeto QMGR en el mandato `setmqaut`.
3. Esto es aplicable si el objeto que debe sustituirse ya existe. Si no existe, la comprobación es como para DEFINE *objeto* NOREPLACE.

Información relacionada

Agrupación en clúster: utilización de las recomendaciones de REFRESH CLUSTER

ALW *Autorizaciones para mandatos PCF*

Esta sección resume las autorizaciones necesarias para cada mandato PCF.

La indicación *No se comprueba* significa que no se lleva a cabo ninguna comprobación de autorización; *No aplicable* significa que esta operación no es relevante para este tipo de objeto.

El ID de usuario bajo el que se ejecuta el programa que envía el mandato también debe tener las autorizaciones siguientes:

- Autorización MQZAO_CONNECT para el gestor de colas
- Autorización MQZAO_DISPLAY sobre el gestor de colas para realizar mandatos PCF

La autorización especial MQZAO_ALL_ADMIN incluye todas las autorizaciones de la lista siguiente que sean relevantes para el tipo de objeto, excepto MQZAO_CREATE, que no es específica de un objeto o tipo de objeto determinado.

Change *objeto*

Objeto	**Autorización necesaria**
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	MQZAO_CHANGE
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE
Información de comunicación	MQZAO_CHANGE

Clear *objeto*

Objeto	**Autorización necesaria**
Cola	MQZAO_CLEAR
Tema	MQZAO_CLEAR
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	No aplicable
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable
Información de comunicación	No aplicable

Copy *objeto* (sin sustitución) (1)

Objeto	Autorización necesaria
Cola	MQZAO_CREATE **(2)**
Tema	MQZAO_CREATE **(2)**
Proceso	MQZAO_CREATE **(2)**
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CREATE **(2)**
Información de autenticación	MQZAO_CREATE **(2)**
Canal	MQZAO_CREATE **(2)**
Canal de conexión de cliente	MQZAO_CREATE **(2)**
Escucha	MQZAO_CREATE **(2)**
Servicio	MQZAO_CREATE **(2)**
Información de comunicación	MQZAO_CREATE **(“2” en la página 153)**

Copiar *objeto* (con sustitución) (1, 4)

Objeto	Autorización necesaria
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE
Información de comunicación	MQZAO_CHANGE

Create *objeto* (sin sustitución) (3)

Objeto	Autorización necesaria
Cola	MQZAO_CREATE **(2)**
Tema	MQZAO_CREATE **(2)**
Proceso	MQZAO_CREATE **(2)**
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CREATE **(2)**
Información de autenticación	MQZAO_CREATE **(2)**
Canal	MQZAO_CREATE **(2)**

Objeto	Autorización necesaria
Canal de conexión de cliente	MQZAO_CREATE **(2)**
Escucha	MQZAO_CREATE **(2)**
Servicio	MQZAO_CREATE **(2)**
Información de comunicación	MQZAO_CREATE **(2)**

Crear *objeto* (con sustitución) (3, 4)

Objeto	Autorización necesaria
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE
Información de comunicación	MQZAO_CHANGE

Delete *objeto*

Objeto	Autorización necesaria
Cola	MQZAO_DELETE
Tema	MQZAO_DELETE
Proceso	MQZAO_DELETE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_DELETE
Información de autenticación	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de conexión de cliente	MQZAO_DELETE
Escucha	MQZAO_DELETE
Servicio	MQZAO_DELETE
Información de comunicación	MQZAO_DELETE

Inquire *objeto*

Objeto	Autorización necesaria
Cola	MQZAO_DISPLAY
Tema	MQZAO_DISPLAY

Objeto	Autorización necesaria
Proceso	MQZAO_DISPLAY
Gestor de colas	MQZAO_DISPLAY
Lista de nombres	MQZAO_DISPLAY
Información de autenticación	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de conexión de cliente	MQZAO_DISPLAY
Escucha	MQZAO_DISPLAY
Servicio	MQZAO_DISPLAY
Información de comunicación	MQZAO_DISPLAY

Inquire *objeto* names

Objeto	Autorización necesaria
Cola	No se comprueba
Tema	No se comprueba
Proceso	No se comprueba
Gestor de colas	No se comprueba
Lista de nombres	No se comprueba
Información de autenticación	No se comprueba
Canal	No se comprueba
Canal de conexión de cliente	No se comprueba
Escucha	No se comprueba
Servicio	No se comprueba
Información de comunicación	No se comprueba

Inicie *objeto*

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	MQZAO_CONTROL
Servicio	MQZAO_CONTROL

Objeto	Autorización necesaria
Información de comunicación	No aplicable

Pare *objeto*

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	MQZAO_CONTROL
Servicio	MQZAO_CONTROL
Información de comunicación	No aplicable

Mandatos de canal

Mandato	Objeto	Autorización necesaria
Sondear canal	Canal	MQZAO_CONTROL
Restablecer canal	Canal	MQZAO_CONTROL_EXTENDED
Resolver canal	Canal	MQZAO_CONTROL_EXTENDED

Mandatos de suscripción

Mandato	Objeto	Autorización necesaria
Cambiar suscripción	Tema	MQZAO_CONTROL
Crear suscripción	Tema	MQZAO_CONTROL
Suprimir suscripción	Tema	MQZAO_CONTROL
Consultar suscripción	Tema	MQZAO_DISPLAY

Mandatos de seguridad

Mandato	Objeto	Autorización necesaria
Establecer registro de autorización	Gestor de colas	MQZAO_CHANGE
Suprimir registro de autorización	Gestor de colas	MQZAO_CHANGE
Consultar registros de autorización	Gestor de colas	MQZAO_DISPLAY
Consultar servicio de autorización	Gestor de colas	MQZAO_DISPLAY

Mandato	Objeto	Autorización necesaria
Consultar autorización de entidad	Gestor de colas	MQZAO_DISPLAY
Establecer registro de autenticación de canal	Gestor de colas	MQZAO_CHANGE
Consultar registros de autenticación de canal	Gestor de colas	MQZAO_DISPLAY
Renovar seguridad	Gestor de colas	MQZAO_CHANGE

Muestra el estado

Mandato	Objeto	Autorización necesaria
Consultar estado del canal	Gestor de colas	MQZAO_DISPLAY Tenga en cuenta que la autorización `+inq` (o equivalente MQZAO_INQUIRE) es necesaria en la cola de transmisión si el tipo de canal es CLUSSDR.
Consultar estado de escucha de canal	Gestor de colas	MQZAO_DISPLAY
Consultar estado de publicación/suscripción	Gestor de colas	MQZAO_DISPLAY
Consultar estado de suscripción	Gestor de colas	MQZAO_DISPLAY
Consultar estado del servicio	Gestor de colas	MQZAO_DISPLAY
Consultar estado de tema	Gestor de colas	MQZAO_DISPLAY

Mandatos de clúster

Mandato	Objeto	Autorización necesaria
Consultar gestor de colas de clúster	Gestor de colas	MQZAO_DISPLAY
Renovar clúster	grupo de pertenencia 'mqm' necesario	grupo de pertenencia 'mqm' necesario
Restablecer clúster	grupo de pertenencia 'mqm' necesario	grupo de pertenencia 'mqm' necesario
Suspender clúster de gestores de colas	grupo de pertenencia 'mqm' necesario	grupo de pertenencia 'mqm' necesario
Reanudar clúster de gestores de colas	grupo de pertenencia 'mqm' necesario	grupo de pertenencia 'mqm' necesario

Otros mandatos administrativos

Mandato	Objeto	Autorización necesaria
Sondear gestor de colas	Gestor de colas	MQZAO_DISPLAY
Renovar gestor de colas	Gestor de colas	MQZAO_CHANGE

Mandato	Objeto	Autorización necesaria
Restablecer gestor de colas	Gestor de colas	MQZAO_CHANGE
Restablecer estadísticas de la cola	Cola	MQZAO_DISPLAY y MQZAO_CHANGE
Consultar conexión	Gestor de colas	MQZAO_DISPLAY
Detener conexión	Gestor de colas	MQZAO_CHANGE

Nota:

1. En los mandatos Copy, también es necesaria la autorización MQZAO_DISPLAY para el objeto de origen.
2. La autorización MQZAO_CREATE no es específica de un objeto o tipo de objeto en particular. Para un gestor de colas especificado, la autorización de creación se otorga para todos los objetos, especificando un tipo de objeto QMGR en el mandato `setmqaut`.
3. Para los mandatos Create, también se necesita la autorización MQZAO_DISPLAY para el SYSTEM.DEFAULT.* .
4. Esto es aplicable si el objeto que debe sustituirse ya existe. Si no existe, la comprobación es como para un mandato Copy o Create sin sustitución.

AIX Creación y gestión de grupos en AIX

En AIX, siempre y cuando no esté utilizando NIS o NIS+, utilice SMITTY para trabajar con grupos.

Acerca de esta tarea

En AIX, puede utilizar SMITTY para crear un grupo, añadir un usuario a un grupo, mostrar una lista de los usuarios que están en el grupo y eliminar un usuario de un grupo.

Procedimiento

1. Desde SMITTY, seleccione **Seguridad y usuarios** y pulse Intro.
2. Seleccione **Grupos** y pulse Intro.
3. Para crear un grupo, realice los pasos siguientes:
   a) Seleccione **Añadir un grupo** y pulse Intro.
   b) Escriba el nombre del grupo y los nombres de los usuarios que desee añadir al grupo, separados por comas.
   c) Pulse Intro para crear el grupo.
4. Para añadir un usuario a un grupo, efectúe los pasos siguientes:
   a) Seleccione **Cambiar/Mostrar características de grupos** y pulse Intro.
   b) Escriba el nombre del grupo para que aparezca una lista de los miembros del grupo.
   c) Añada los nombres de los usuarios que desea añadir al grupo, separados por comas.
   d) Pulse Intro para añadir los nombres al grupo.
5. Para mostrar quién está en un grupo, realice los pasos siguientes:
   a) Seleccione **Cambiar/Mostrar características de grupos** y pulse Intro.
   b) Escriba el nombre del grupo para que aparezca una lista de los miembros del grupo.
6. Para eliminar un usuario de un grupo, realice los pasos siguientes:
   a) Seleccione **Cambiar/Mostrar características de grupos** y pulse Intro.
   b) Escriba el nombre del grupo para que aparezca una lista de los miembros del grupo.
   c) Suprima el nombre del usuario que desea eliminar del grupo.
   d) Pulse Intro para eliminar el nombre del grupo.

Linux Creación y gestión de grupos en Linux

En Linux, siempre y cuando no esté utilizando NIS o NIS+, utilice el archivo `/etc/group` para trabajar con grupos.

Acerca de esta tarea

En Linux, la información de grupo se mantiene en el archivo `/etc/group`. Puede utilizar mandatos para crear grupos, añadir usuarios a grupos, visualizar una lista de los usuarios que están en un grupo y eliminar usuarios de un grupo.

Procedimiento

1. Para crear un nuevo grupo, utilice el mandato **groupadd**.
   Escriba el siguiente mandato:

   ```
   groupadd -g group-ID group-name
   ```

   donde *ID_grupo* es el identificador numérico del grupo y *nombre_grupo* es el nombre del grupo.
2. Para añadir un miembro a un grupo adicional, utilice el mandato **usermod** que enumera los grupos adicionales de los que el usuario es miembro actualmente y los grupos adicionales de los que el usuario va a ser miembro.
   Por ejemplo, si el usuario ya es miembro del grupo `groupa` y va a convertirse en miembro de `groupb`, utilice el mandato siguiente:

   ```
   usermod -G groupa,groupb user-name
   ```

   donde *nombre_usuario* es el nombre de usuario.
3. Para visualizar los miembros de un grupo, utilice el mandato **getent**.
   Escriba el siguiente mandato:

   ```
   getent group group-name
   ```

   donde *nombre-grupo* es el nombre del grupo.
4. Para eliminar un miembro de un grupo adicional, utilice el mandato **usermod**, que enumera los grupos adicionales de los que desea que el usuario siga siendo miembro.
   Por ejemplo, si el grupo primario del usuario es `users` y el usuario también es miembro de los grupos `mqm`, `groupa` y `groupb`, para eliminar el usuario del grupo `mqm`, utilice el mandato siguiente:

   ```
   usermod -G groupa,groupb user-name
   ```

   donde *nombre_usuario* es el nombre de usuario.

Windows Creación y gestión de grupos en Windows

En Windows, utilice la característica Administración de equipos para administrar los grupos en una estación de trabajo o en una máquina del servidor miembro.

Acerca de esta tarea

Para los controladores de dominio, los usuarios y grupos se administran mediante Active Directory. Para obtener más información sobre la utilización de Active Directory, consulte las instrucciones correspondientes del sistema operativo.

Los cambios que realice en la pertenencia a un grupo principal no se reconocen hasta que se reinicia el gestor de colas o se emite el mandato MQSC **REFRESH SECURITY** (o el equivalente PCF).

Utilice el panel Administración de equipos de Windows para trabajar con el usuario y los grupos. Puede que los cambios efectuados en la sesión iniciada actual no sean efectivos hasta que se vuelva a iniciar la sesión.

Windows *Creación de un grupo en Windows*

Crear un grupo utilizando el panel de control.

Procedimiento

1. Abra el Panel de control
2. Efectúe una doble pulsación en **Herramientas administrativas**.
   Se abre el panel Herramientas administrativas.
3. Efectúe una doble pulsación en **Administración de equipos**.
   Se abre el panel Administración de equipos.
4. Expanda **Usuarios locales y grupos**.
5. Pulse el botón derecho del ratón en **Grupos** y seleccione **Grupo nuevo...**
   Aparece el panel Grupo nuevo.
6. Escriba un nombre adecuado en el campo Nombre de grupo y pulse **Crear**.
7. Pulse **Cerrar**.

Windows *Adición de un usuario a un grupo en Windows*

Añada un usuario a un grupo utilizando el panel de control.

Procedimiento

1. Abra el Panel de control
2. Efectúe una doble pulsación en **Herramientas administrativas**.
   Se abre el panel Herramientas administrativas.
3. Efectúe una doble pulsación en **Administración de equipos**.
   Se abre el panel Administración de equipos.
4. Desde el panel Administración de equipos, expanda **Usuarios locales y grupos**.
5. Seleccione **Usuarios**
6. Efectúe una doble pulsación en el usuario que desea añadir a un grupo.
   Aparece el panel de propiedades de usuario.
7. Seleccione el separador **Miembro de**.
8. Seleccione el grupo al que desea añadir el usuario. Si el grupo que desea no está visible:
   a) Pulse **Añadir...**.
      Aparece el panel Seleccionar grupos.
   b) Pulse **Ubicaciones...**.
      Aparece el panel Ubicaciones.
   c) Seleccione la ubicación del grupo al que desea añadir el usuario en la lista y pulse **Aceptar**.
   d) Escriba el nombre de grupo en el campo correspondiente.

      De forma alternativa, pulse **Avanzado ...** y, a continuación, **Buscar ahora** para listar los grupos disponibles en la ubicación seleccionada actualmente. Aquí, seleccione el grupo al que desea añadir el usuario y pulse **Aceptar**.
   e) Pulse **Aceptar**.
      Aparece el panel de propiedades de usuario, que muestra el grupo que ha añadido.
   f) Seleccione el grupo.
9. Pulse **Aceptar**.

Aparece el panel Administración de equipos.

Windows *Visualización de los miembros de un grupo en Windows*

Mostrar los miembros de un grupo utilizando el panel de control.

Procedimiento

1. Abra el Panel de control
2. Efectúe una doble pulsación en **Herramientas administrativas**.
   Se abre el panel Herramientas administrativas.
3. Efectúe una doble pulsación en **Administración de equipos**.
   Se abre el panel Administración de equipos.
4. Desde el panel Administración de equipos, expanda **Usuarios locales y grupos**.
5. Seleccione **Grupos**.
6. Efectúe una doble pulsación en un grupo. Aparece el panel de propiedades del grupo.
   Aparece el panel de propiedades del grupo.

Resultados

Se muestran los miembros del grupo.

Windows *Supresión de un usuario de un grupo en Windows*

Eliminar un usuario de un grupo utilizando el panel de control.

Procedimiento

1. Abra el Panel de control
2. Efectúe una doble pulsación en **Herramientas administrativas**.
   Se abre el panel Herramientas administrativas.
3. Efectúe una doble pulsación en **Administración de equipos**.
   Se abre el panel Administración de equipos.
4. Desde el panel Administración de equipos, expanda **Usuarios locales y grupos**.
5. Seleccione **Usuarios**.
6. Efectúe una doble pulsación en el usuario que desea añadir a un grupo.
   Aparece el panel de propiedades de usuario.
7. Seleccione el separador **Miembro de**.
8. Seleccione el grupo del que desea eliminar el usuario y luego pulse **Quitar**.
9. Pulse **Aceptar**.
   Aparece el panel Administración de equipos.

Resultados

Ya ha eliminado al usuario del grupo.

Windows Consideraciones especiales de seguridad en Windows

Algunas funciones de seguridad se comportan de forma diferente en las distintas versiones de Windows.

La seguridad de IBM MQ depende de llamadas a la API del sistema operativo para obtener información sobre autorizaciones de usuario y pertenencias a grupos. Algunas funciones no se comportan del mismo modo en los sistemas Windows. Esta colección de temas incluye descripciones de cómo estas diferencias pueden afectar a la seguridad de IBM MQ cuando se ejecuta IBM MQ en un entorno Windows.

Windows *Cuentas de usuario local y de dominio para el servicio IBM MQ Windows*

Cuando IBM MQ está en ejecución, debe comprobar que sólo los usuarios autorizados pueden acceder a los gestores de colas o a las colas. Esto requiere una cuenta de usuario especial que IBM MQ puede utilizar para consultar información sobre el usuario que intenta dicho acceso.

- "Configuración de cuentas de usuario especiales con el Prepare IBM MQ Wizard" en la página 157
- "Utilización de IBM MQ con Active Directory" en la página 157
- "Derechos de usuario necesarios para un servicio IBM MQ Windows" en la página 158

Configuración de cuentas de usuario especiales con el Prepare IBM MQ Wizard

El Prepare IBM MQ Wizard crea una cuenta de usuario especial para que el servicio Windows pueda ser compartido por los procesos que necesitan utilizarlo (consulte Configuración de IBM MQ con el PPrepare IBM MQ Wizard).

Un servicio de Windows se comparte entre procesos de cliente para una instalación de IBM MQ. Se crea un servicio para cada instalación. Cada servicio se denomina `MQ_`*`InstallationName`*y tiene un nombre de visualización de `IBM MQ(`*`InstallationName`*`)`.

Puesto que cada servicio se debe compartir entre sesiones de inicio de sesión interactivas y no interactivas, debe iniciar cada una bajo una cuenta de usuario especial. Puede utilizar una cuenta de usuario especial para todos los servicios o crear distintas cuentas de usuario especiales. Cada cuenta de usuario especial debe tener el derecho de usuario `Iniciar sesión como servicio`; para obtener más información, consulte Tabla 14 en la página 158. Si el ID de usuario no tiene autorización para poder ejecutar el servicio, el servicio no se inicia y se devuelve un error al registro de sucesos del sistema de Windows. Normalmente, tendrá que ejecutar Prepare IBM MQ Wizard y configurar el ID de usuario correctamente. Sin embargo, si ha configurado el ID de usuario manualmente, es posible que pueda tener un problema que será necesario resolver.

Cuando instala IBM MQ y ejecuta el Prepare IBM MQ Wizard por primera vez, se crea una cuenta de usuario local para el servicio denominado MUSR_MQADMIN con los valores y permisos necesarios, incluido `Iniciar sesión como un servicio`.

En instalaciones posteriores, el Prepare IBM MQ Wizard creará una cuenta de usuario denominada MUSR_MQADMIN*x*, donde *x* es el siguiente número disponible que representa un ID de usuario que no existe. La contraseña para MUSR_MQADMIN*x* se genera aleatoriamente cuando se crea la cuenta y se utiliza para configurar el entorno de inicio de sesión para el servicio. La contraseña generada no caduca.

Esta cuenta de IBM MQ MQ no se ve afectada por ninguna de las políticas de cuentas definidas en el sistema que requieren que las contraseñas se cambien después de un cierto periodo de tiempo.

La contraseña no se conoce fuera de este proceso de un solo uso y la almacena el sistema operativo de Windows en una parte segura del registro.

Utilización de IBM MQ con Active Directory

En algunas configuraciones de red, donde las cuentas de usuario se definen en controladores de dominios que están utilizando el servicio de directorios Active Directory, la cuenta de usuario local bajo la cual se está ejecutando IBM MQ podría no tener la autorización que requiere para consultar la pertenencia de otras cuentas de usuario de dominio. Cuando instala IBM MQ, el Prepare IBM MQ Wizard identifica si este es el caso realizando pruebas y formulando preguntas sobre la configuración de red.

Si la cuenta de usuario local con la que se ejecuta IBM MQ no tiene la autorización necesaria, el Prepare IBM MQ Wizard le solicita los detalles de la cuenta de usuario de dominio con derechos de usuario concretos. Para obtener información acerca de cómo crear y configurar una cuenta de dominio de Windows, consulte Creación y configuración de cuentas de dominio de Windows para IBM MQ. Para ver los derechos de usuario que necesita la cuenta de usuario de dominio, consulte Tabla 14 en la página 158.

Cuando ha especificado los detalles de cuenta válidos para la cuenta de usuario de dominio en el Prepare IBM MQ Wizard, éste último configura un servicio de IBM MQWindows para que se ejecute con la nueva

cuenta. Los detalles de la cuenta se guardan en una parte segura del Registro que no pueden leer los usuarios.

Cuando el servicio se ejecuta, se inicia un servicio IBM MQ Windows y permanece en ejecución mientras se ejecuta el servicio. Un administrador de IBM MQ que inicie la sesión en el servidor después de haber lanzado el servicio de Windows puede utilizar IBM MQ Explorer para administrar gestores de colas en el servidor. Esto conecta IBM MQ Explorer al proceso del servicio de Windows existente. Estas dos acciones necesitan niveles de permiso diferentes para poder funcionar:

- El proceso de inicio necesita un permiso de inicio.
- El administrador de IBM MQ requiere permiso de acceso.

Derechos de usuario necesarios para un servicio IBM MQ Windows

En la tabla siguiente se muestran los derechos de usuario necesarios para las cuentas de usuario local y de dominio con las que se ejecuta el servicio Windows para una instalación de IBM MQ.

Tabla 14. Derechos de usuario necesarios para un servicio Windows de IBM MQ

Permiso	**Descripción**
Iniciar sesión como proceso por lotes	Permite que un servicio IBM MQ Windows se ejecute en esta cuenta de usuario.
Iniciar sesión como servicio	Permite a los usuarios establecer el servicio IBM MQ Windows para iniciar sesión utilizando la cuenta configurada.
Concluir el sistema	Permite al servicio IBM MQ Windows reiniciar el servidor si está configurado para ello cuando falla la recuperación de un servicio.
Aumentar cuotas	Necesario para la llamada `CreateProcessAsUser` del sistema operativo.
Actuar como parte del sistema operativo	Necesario para la llamada `LogonUser` del sistema operativo.
Eludir la comprobación cruzada	Necesario para la llamada `LogonUser` del sistema operativo.
Sustituir una señal de nivel de proceso	Necesario para la llamada `LogonUser` del sistema operativo.

Nota: Es posible que se necesiten derechos para depurar programas en entornos que ejecutan aplicaciones ASP e IIS.

Su cuenta de usuario de dominio debe tener estos derechos de usuario de Windows establecidos como derechos de usuario efectivos, tal como se indica en la aplicación Directiva de seguridad local. Si no lo están, establézcalos utilizando la aplicación Directiva de seguridad local localmente en el servidor, o utilizando la aplicación Directiva de seguridad de dominio para todo el dominio.

Windows *Permisos de seguridad de Windows Server*

La instalación de IBM MQ tiene un comportamiento distinto en Windows Server según si la instalación la realiza un usuario local o un usuario de dominio.

Si un usuario *local* instala IBM MQ, el Prepare IBM MQ Wizard detecta que el usuario local creado para el servicio IBM MQ Windows puede recuperar la información de pertenencia a grupos del usuario que realiza la instalación. El Prepare IBM MQ Wizard solicita al usuario información sobre la configuración de red para determinar si hay otras cuentas de usuario definidas o no en los controladores de dominio que se ejecutan en Windows 2000 o posterior. De esta forma, el servicio de IBM MQ Windows se debe ejecutar bajo una cuenta de usuario de dominio con autoridades y valores particulares. El Prepare IBM MQ Wizard

solicita al usuario los detalles de la cuenta de este usuario tal como se describe en Configuración de IBM MQ con el Prepare IBM MQ Wizard.

Si un usuario *domain* instala IBM MQ, el Prepare IBM MQ Wizard detecta que el usuario local creado para el servicio IBM MQ Windows no puede recuperar la información de pertenencia a grupos del usuario que realiza la instalación. En este caso, el Prepare IBM MQ Wizard siempre solicita al usuario los detalles de la cuenta de usuario de dominio para que el servicio IBM MQ Windows los utilice.

Cuando el servicio IBM MQ Windows debe utilizar una cuenta de usuario de dominio, IBM MQ no puede operar correctamente hasta que esto se haya configurado utilizando el Prepare IBM MQ Wizard. El Prepare IBM MQ Wizard no permite que el usuario siga con otras tareas, hasta que no se haya configurado el servicio Windows con una cuenta adecuada.

Para obtener más información, consulte Creación y configuración de cuentas de dominio para IBM MQ.

Windows *Cambio del nombre de usuario asociado al servicio IBM MQ*

Para cambiar el nombre de usuario asociado al servicio IBM MQ cree una cuenta nueva y especifique los detalles mediante el Prepare IBM MQ Wizard.

Acerca de esta tarea

Cuando instala IBM MQ y ejecuta el Prepare IBM MQ Wizard por primera vez, se crea una cuenta de usuario local para el servicio denominada MUSR_MQADMIN. En instalaciones posteriores, el Prepare IBM MQ Wizard creará una cuenta de usuario denominada MUSR_MQADMIN*x*, donde *x* es el siguiente número disponible que representa un ID de usuario que no existe.

Es posible que tenga que cambiar el nombre de usuario asociado al servicio IBM MQ, MUSR_MQADMIN o MUSR_MQADMIN*x*, por algún otro. Por ejemplo, es posible que tenga que hacer esto si el gestor de colas está asociado a Db2, que no acepta nombres de usuario de más de 8 caracteres.

Procedimiento

1. Cree una nueva cuenta de usuario (por ejemplo, **NEW_NAME**)
2. Utilice el Prepare IBM MQ Wizard para especificar los detalles de la nueva cuenta de usuario.

Tareas relacionadas

Configuración de IBM MQ con Prepare IBM MQ Wizard

Windows *Cambiar la contraseña de la cuenta de usuario local del servicio IBM MQ Windows*

Puede cambiar la contraseña de la cuenta de usuario local del servicio de IBM MQ Windows utilizando el panel Administración de equipos.

Acerca de esta tarea

Para cambiar la contraseña de la cuenta de usuario local de servicio de IBM MQWindows , realice los pasos siguientes:

Procedimiento

1. Identifique el usuario en el que se ejecuta el servicio.
2. Detenga el servicio IBM MQ desde el panel Administración de equipos.
3. Cambie la contraseña necesaria igual que lo haría con una contraseña personal.
4. Vaya a las propiedades del servicio IBM MQ desde el panel Administración de equipos.
5. Seleccione la página **Iniciar sesión**.
6. Confirme que el nombre de cuenta especificado coincida con el usuario para el que se ha modificado la contraseña.
7. Escriba la contraseña en los campos **Contraseña** y **Confirmar contraseña** y pulse **Aceptar**.

Windows *Cambio de la contraseña de un servicio IBM MQ Windows para una instalación que se ejecuta con una cuenta de usuario de dominio*

Como alternativa a la utilización del Prepare IBM MQ Wizard para especificar los detalles de la cuenta de usuario de dominio, puede utilizar el panel Administración de equipos para modificar los detalles de **Iniciar sesión** para el servicio IBM MQ específico de la instalación.

Acerca de esta tarea

Si el servicio de IBM MQWindows para una instalación se está ejecutando bajo una cuenta de usuario de dominio, puede cambiar la contraseña de la cuenta de la siguiente manera:

Procedimiento

1. Cambie la contraseña de la cuenta de dominio en el controlador de dominio. Es posible que debe pedirle al administrador del sistema que realice esta tarea.
2. Complete los pasos siguientes para modificar la página **Iniciar sesión** para el servicio IBM MQ.
   a) Identifique el usuario con el que se ejecuta el servicio.
   b) Detenga el servicio IBM MQ desde el panel Administración de equipos.
   c) Cambie la contraseña necesaria igual que lo haría con una contraseña personal.
   d) Vaya a las propiedades del servicio IBM MQ desde el panel Administración de equipos.
   e) Seleccione la página **Iniciar sesión**.
   f) Confirme que el nombre de cuenta especificado coincida con el usuario para el que se ha modificado la contraseña.
   g) Escriba la contraseña en los campos **Contraseña** y **Confirmar contraseña** y pulse **Aceptar**.

   La cuenta de usuario bajo la cual se ejecute el servicio IBM MQ Windows ejecuta los mandatos MQSC que han emitido las aplicaciones de la interfaz de usuario, o que se han realizado automáticamente durante el arranque del sistema, la conclusión o la recuperación del servicio. Por lo tanto, esta cuenta de usuario debe tener derechos de administración de IBM MQ. De forma predeterminada, se añade al grupo mqm local en el servidor. Si se elimina esta pertenencia, el servicio IBM MQ Windows no funciona. Para obtener más información sobre los derechos de usuario, consulte “Derechos de usuario necesarios para un servicio IBM MQ Windows” en la página 158.

   Si surge un problema de seguridad con la cuenta de usuario bajo la que se ejecuta el servicio IBM MQ Windows, aparecerán mensajes de error y descripciones en la anotación de sucesos del sistema.

Tareas relacionadas

Configuración de IBM MQ con Prepare IBM MQ Wizard

Windows ***Consideraciones sobre la promoción de servidores Windows en los controladores de dominio***

Cuando promueve un servidor Windows a un controlador de dominio, considere si el valor de seguridad relacionado con los permisos de usuario y grupo es adecuado. Al cambiar el estado de una máquina de Windows entre el servidor y el controlador de dominio, tenga en cuenta que esto puede afectar al funcionamiento de IBM MQ porque IBM MQ utiliza un grupo mqm definido localmente.

Valores de seguridad relacionados con los permisos de usuario y grupo de dominio

IBM MQ se basa en la información de pertenencia a grupos para implementar su política de seguridad, lo que significa que es importante que el ID de usuario que realiza operaciones de IBM MQ pueda determinar la pertenencia a grupos de otros usuarios.

Cuando promueve un servidor Windows a un controlador de dominio, se le presenta una opción para el valor de seguridad relacionado con los permisos de usuario y grupo. Esta opción controla si los usuarios arbitrarios pueden recuperar miembros de grupo desde Active Directory. Si se configura un controlador de dominio para que las cuentas locales tengan autorización para consultar la pertenencia

a grupos de las cuentas de usuario de dominio, el ID de usuario predeterminado creado por IBM MQ durante el proceso de instalación puede obtener la pertenencia a grupos de otros usuarios según sea necesario. Sin embargo, si se configura un controlador de dominio de modo que las cuentas locales no tengan autorización para consultar la pertenencia a grupos de las cuentas de usuario de dominio, esto impide que IBM MQ complete sus comprobaciones de que los usuarios definidos en el dominio tengan autorización para acceder a los gestores de colas o las colas, y el acceso falla. Si está utilizando Windows en un controlador de dominio que se ha configurado de esta forma, se debe utilizar una cuenta de usuario de dominio especial con los permisos necesarios.

En este caso, debe saber:

- Cómo se comportan los permisos de seguridad para la versión de Windows.
- Cómo permitir que los miembros del grupo mqm de dominio lean la información de pertenencia a grupos.
- Cómo configurar un servicio de IBM MQWindows para que se ejecute bajo un usuario de dominio.

Para obtener más información, consulte Configuración de cuentas de usuario de IBM MQ.

Acceso de IBM MQ al grupo mqm local

Cuando los servidores Windows se promocionan o se degradan en controladores de dominio, IBM MQ pierde el acceso al grupo mqm local.

Cuando un servidor se promociona para que sea un controlador de dominio, el ámbito cambia de local a local del dominio. Cuando la máquina se degrada a servidor, todos los grupos locales del dominio se eliminan. Esto significa que cuando una máquina pasa de servidor a controlador de dominio y luego vuelve al estado de servidor pierde el acceso a un grupo mqm local. El síntoma es un error que indica que falta un grupo mqm local, por ejemplo:

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

Para solucionar este problema, vuelva a crear el grupo mqm local utilizando las herramientas de gestión estándares de Windows. Puesto que toda la información de pertenencia a grupos se pierde, debe volver a incluir los usuarios de IBM MQ con privilegios en el grupo mqm local que acaba de crear. Si la máquina es un miembro del dominio, también debe añadir el grupo mqm de dominio al grupo mqm local, para otorgar a los ID de usuario IBM MQ de dominio con privilegios el nivel de autorización necesario.

Windows *Restricciones en los grupos anidados en Windows*

Existen restricciones en el uso de grupos anidados. Estas restricciones se deben en parte al nivel funcional del dominio y en parte a restricciones de IBM MQ.

Active Directory puede dar soporte a distintos tipos de grupos en un contexto de dominio dependiendo del nivel funcional del dominio. De forma predeterminada, los dominios de Windows 2003 están en el directorio " Windows 2000 mixed " nivel funcional. (Windows Server 2008 y Windows Server 2012 siguen el modelo de dominio Windows 2003 .) El nivel funcional del dominio determina los tipos de grupos soportados y el nivel de anidamiento permitido al configurar los ID de usuario en un entorno de dominio. Consulte la documentación de Active Directory para obtener información detallada sobre el Ámbito de grupo y los criterios de inclusión.

Además de los requisitos de Active Directory, se imponen restricciones adicionales para los ID utilizados por IBM MQ. Las API de red que utiliza IBM MQ no dan soporte a todas las configuraciones a las que da soporte el nivel funcional del dominio. Como resultado, IBM MQ no puede consultar la pertenencia a grupos de cualquier ID de dominio presente en un grupo Local de dominio que luego se anida en un grupo local. Además, no se da soporte al anidamiento múltiple de grupos globales y universales. No obstante, los grupos globales o universales anidados inmediatamente están soportados.

Windows *Autorización de usuarios para utilizar IBM MQ de forma remota*

Si tiene que crear e iniciar gestores de colas cuando esté conectado a IBM MQ de forma remota, debe tener el acceso de usuario `Crear objetos globales`.

Acerca de esta tarea

Nota: Los administradores tienen el acceso de usuario `Crear objetos globales` de forma predeterminada, así pues si usted es un administrador, puede crear e iniciar los gestores de colas cuando están conectados de forma remota sin alterar los derechos de usuario.

Si se está conectando a una máquina de Windows utilizando Terminal Services o una conexión de escritorio remoto y tiene problemas para crear, iniciar o suprimir un gestor de colas, esto puede ser debido a que no tiene el acceso de usuario `Crear objetos globales`.

El acceso de usuario `Crear objetos globales` limita a los usuarios autorizados a crear objetos en el espacio de nombres globales. Para que una aplicación pueda crear un objeto global, se debe ejecutar en el espacio de nombres global o el usuario en el que se está ejecutando la aplicación debe disponer del acceso de usuario `Crear objetos globales`.

Cuando se conecta de forma remota a una máquina Windows utilizando Terminal Services o una conexión de escritorio remoto, las aplicaciones se ejecutan en su propio espacio de nombres local. Si intenta crear o suprimir un gestor de colas utilizando IBM MQ Explorer o el mandato **crtmqm** o **dltmqm**, o iniciar un gestor de colas utilizando el mandato **strmqm**, se genera un error de autorización. Así se crea un FDC de IBM MQ con el ID de analizador XY132002.

El inicio de un gestor de colas utilizando IBM MQ Explorer o utilizando el mandato **amqmdain qmgr start** funciona correctamente porque estos mandatos no inician directamente el gestor de colas. En cambio, los mandatos envían la solicitud para iniciar el gestor de colas a un proceso independiente que se está ejecutando en el espacio de nombres global.

Si los distintos métodos de administración de IBM MQ no funcionan cuando se utiliza Terminal Services, intente establecer el derecho de usuario `Crear objetos globales`.

Procedimiento

1. Abra el panel Herramientas administrativas:

   Windows Server 2008 y Windows Server 2012
   Acceda a este panel utilizando **Panel de control** > **Sistema y mantenimiento** > **Herramientas administrativas**.

   Windows 8.1
   Acceda a este panel utilizando **Herramientas administrativas** > **Administración de equipos**
2. Efectúe una doble pulsación en **Directiva de seguridad local**.
3. Expanda **Directivas locales**.
4. Pulse **Asignación de derechos de usuario**.
5. Añada el usuario o grupo nuevo a la directiva `Crear objetos globales`.

Windows *El programa de salida de canal SSPI en Windows*

IBM MQ for Windows proporciona un programa de salida de seguridad, que se puede utilizar tanto en canales de mensajes como en canales MQI. La salida se suministra como código fuente y código objeto, y proporciona autenticación unidireccional y bidireccional.

La salida de seguridad utiliza la Interfaz del proveedor de soporte para seguridad (SSPI), que proporciona los recursos de seguridad integrados de las plataformas Windows.

La salida de seguridad proporciona los siguientes servicios de identificación y autenticación:

Autenticación unidireccional
Esto utiliza el soporte de autenticación de Windows NT LAN Manager (NTLM). NTLM permite a los servidores autenticar sus clientes. No permite que un cliente autentique un servidor, ni que un servidor autentique otro. NTLM se ha diseñado para un entorno de red en el que se da por supuesto que los servidores son genuinos. NTLM está soportado en todas las plataformas Windows soportadas en IBM WebSphere MQ 7.0.

Este servicio se suele utilizar en un canal MQI para permitir que un gestor de colas del servidor autentique una aplicación IBM MQ MQI client. Una aplicación cliente se identifica mediante el ID de usuario asociado con el proceso que se está ejecutando.

Para llevar a cabo la autenticación, la salida de seguridad en el extremo cliente de un canal adquiere una señal de autenticación de NTLM y envía la señal en un mensaje de seguridad a su asociado en el otro extremo del canal. La salida de seguridad del asociado pasa la señal a NTLM, el cual comprueba que la señal es auténtica. Si la salida de seguridad del asociado no está satisfecha con la autenticidad de la señal, indica al MCA que cierre el canal.

Autenticación bidireccional o mutua

Utiliza los servicios de autenticación de Kerberos. El protocolo Kerberos no da por supuesto que los servidores de un entorno de red son genuinos. Los servidores pueden autenticar clientes y otros servidores, y los clientes pueden autenticar servidores. Kerberos está soportado en todas las plataformas Windows soportadas en IBM WebSphere MQ 7.0.

Este servicio se puede utilizar en canales de mensajes y MQI. En un canal de mensajes, proporciona autenticación mutua de los dos gestores de colas. En un canal MQI, permite que el gestor de colas del servidor y la aplicación IBM MQ MQI client se autentiquen entre sí. Un gestor de colas se identifica por su nombre con el prefijo de la serie `ibmMQSeries/`. Una aplicación cliente se identifica mediante el ID de usuario asociado con el proceso que se está ejecutando.

Para realizar la autenticación mutua, la salida de seguridad inicial adquiere una señal de autenticación del servidor de seguridad Kerberos y envía la señal en un mensaje de seguridad a su asociado. La salida de seguridad del asociado pasa la señal al servidor de seguridad Kerberos, el cual comprueba que es auténtica. El servidor de seguridad Kerberos genera una segunda señal, que el asociado envía en un mensaje de seguridad a la salida de seguridad inicial. La salida de seguridad inicial solicita al servidor Kerberos que compruebe que la segunda señal es auténtica. Durante este intercambio, si alguna de las salidas de seguridad no está satisfecha con la autenticidad de la señal enviada por la otra, indica al MCA que cierre el canal.

La salida de seguridad se suministra en formato fuente y objeto. Puede utilizar el código fuente como punto de partida para escribir sus propios programas de salida de canal o puede utilizar el módulo objeto tal como se suministra. El módulo objeto tiene dos puntos de entrada, uno para la autenticación unidireccional mediante el soporte para autenticación NTLM y el otro para la autenticación bidireccional mediante servicios de autenticación de Kerberos.

Para obtener más información sobre cómo funciona el programa de salida de canal SSPI y para ver instrucciones sobre cómo implementarlo, consulte Utilización de la salida de seguridad SSPI en sistemas Windows.

Windows *Aplicación de archivos de plantilla de seguridad en Windows*

La aplicación de una plantilla puede afectar a los valores de seguridad aplicados a los archivos y directorios de IBM MQ. Si utiliza la plantilla de alta seguridad, aplíquela antes de instalar IBM MQ.

Windows soporta archivos de plantilla de seguridad basados en texto que puede utilizar para aplicar valores de seguridad uniformes a uno o más sistemas con el complemento Configuración y análisis de seguridad de MMC. En particular, Windows proporciona varias plantillas que incluyen un rango de valores de seguridad con objeto de proporcionar niveles de seguridad específicos. Estas plantillas de seguridad predefinidas incluyen las plantillas Compatible, Segura y De alta seguridad.

La aplicación de una de estas plantillas puede afectar a los valores de seguridad aplicados a los archivos y directorios de IBM MQ. Si desea utilizar la plantilla De alta seguridad, configure la máquina antes de instalar IBM MQ.

Si aplica la plantilla de alta seguridad en una máquina en la que IBM MQ ya está instalado, todos los permisos que haya establecido en los archivos y directorios de IBM MQ se eliminarán. Puesto que estos permisos se eliminan, perderá el acceso al grupo *Administradores*, *mqm*, y, si procede, el acceso al grupo *Todos* desde los directorios de error.

Windows *Configuración de autorización adicional para aplicaciones Windows que se conectan a IBM MQ*

Es posible que la cuenta con la que se ejecutan los procesos de IBM MQ requiera autorización adicional antes de que se pueda otorgar acceso SYNCHRONIZE a los procesos de aplicaciones.

Acerca de esta tarea

Es posible que experimente problemas si tiene aplicaciones Windows, por ejemplo páginas ASP, que se conectan a IBM MQ y que están configuradas para ejecutarse a un nivel de seguridad superior al habitual.

IBM MQ requiere acceso SYNCHRONIZE para los procesos de aplicaciones a fin de coordinar ciertas acciones. Cuando una aplicación de servidor intenta por primera vez conectarse a un gestor de colas de IBM MQ modificará el acceso para otorgar autorización SYNCHRONIZE para administradores de IBM MQ. Sin embargo, es posible que la cuenta bajo la que se ejecutan los procesos de IBM MQ necesite autorización adicional antes de que se pueda otorgar el acceso solicitado.

Para configurar autorización adicional para el ID de usuario bajo el que se ejecutan los procesos de IBM MQ, realice los pasos siguientes:

Procedimiento

1. Inicie la herramienta Directiva de seguridad local, pulse **Configuración de seguridad**->**Directivas locales**->**Asignación de derechos de usuario** y pulse **Depurar programas**.
2. Efectúe una doble pulsación en **Depurar programas** y, a continuación, añada el ID de usuario de IBM MQ a la lista

   Si el sistema está en un dominio Windows y el valor de directiva efectivo no está definido todavía, aunque el valor de directiva local esté definido, el ID de usuario debe autorizarse de la misma manera a nivel de dominio, utilizando la herramienta Directiva de seguridad de dominio.

IBM i Configuración de la seguridad en IBM i

La seguridad para IBM i se implementa utilizando el Gestor de autorizaciones sobre objetos (OAM) de IBM MQ y la seguridad a nivel de objeto de IBM i.

Consideraciones sobre seguridad que deben tenerse presentes al determinar la autorización de acceso a los objetos de IBM MQ.

Debe tener en cuenta las siguientes cuestiones cuando vaya a configurar las autorizaciones de los usuarios de la empresa:

1. Otorgue y revoque autorizaciones para los mandatos de IBM MQ for IBM i mediante los mandatos IBM i GRTOBJAUT y RVKOBJAUT.

   En la biblioteca QMQM, ciertos objetos no de mandato (*cmd) están definidos para tener la autorización ***PUBLIC** establecida en ***USE**. No cambie las autorizaciones de estos objetos ni utilice una lista de autorizaciones para otorgar autorización. Una autorización incorrecta puede comprometer la funcionalidad de IBM MQ.
2. Durante la instalación de IBM MQ for IBM i, se crean los siguientes perfiles de usuario especiales:

   QMQM
   Se utiliza principalmente para funciones internas sólo del producto. No obstante, se puede utilizar para ejecutar aplicaciones de confianza con MQCNO_FASTPATH_BINDINGS. Consulte Conectarse a un gestor de colas mediante la llamada MQCONNX.

   QMQMADM
   Se utiliza como perfil de grupo para los administradores de IBM MQ. El perfil de grupo otorga acceso a los mandatos CL y a los recursos de IBM MQ.

   Si se utiliza SBMJOB para someter programas que llaman a mandatos IBM MQ, USER no debe establecerse explícitamente en QMQMADM. En su lugar, establezca USER en QMQM u otro perfil de usuario que tenga QMQMADM especificado como grupo.

3. Si va a enviar mandatos de canal a gestores de colas remotos, asegúrese de que su perfil de usuario es miembro del grupo QMQMADM en el sistema de destino. Para obtener una lista de los mandatos de canal PCF y MQSC, consulte Mandatos CL de IBM MQ for IBM i.
4. El conjunto de grupos asociado a un usuario se almacena en memoria caché cuando el OAM calcula las autorizaciones de grupo.

   Todos los cambios realizados en la pertenencia a grupos de un usuario después de que el conjunto de grupos se ha almacenado en la memoria caché no se reconocerán hasta que se reinicie el gestor de colas o se ejecute RFRMQMAUT para renovar la seguridad.
5. Limite el número de usuarios que poseen autorización para trabajar con los mandatos que sean especialmente delicados. Entre ellos se encuentran los siguientes mandatos:
   - Creación de un gestor de colas de mensajes (`CRTMQM`)
   - Borrado de un gestor de colas de mensajes (`DLTMQM`)
   - Inicio de un gestor de colas de mensajes (`STRMQM`)
   - Terminación de un gestor de colas de mensajes (`ENDMQM`)
   - Inicio de un servidor de mandatos (`STRMQMCSVR`)
   - Terminación de un servidor de mandatos (`ENDMQMCSVR`)
6. Las definiciones de canal contienen una especificación de programa de salida de seguridad. Requieren consideraciones especiales la creación y la modificación de canales. Encontrará información detallada sobre las salidas de seguridad en “Visión general de las salidas de seguridad” en la página 116.
7. Se pueden sustituir los programas de salida de canal y de supervisor desencadenante. Ha de ser el programador quien se encargue de la seguridad de estas sustituciones.

IBM i Gestor de autorizaciones sobre objetos (OAM) en IBM i

El gestor de autorizaciones sobre objetos (OAM) gestiona las autorizaciones de los usuarios para manipular objetos de IBM MQ, incluyendo colas y definiciones de proceso. También proporciona una interfaz de mandatos mediante la cual puede otorgar o revocar la autorización de acceso a un objeto para un grupo de usuarios específico. La decisión de permitir el acceso a un recurso la toma el OAM, y el gestor de colas actúa según la decisión tomada. Si el OAM no puede tomar una decisión, el gestor de colas impide el acceso a dicho recurso.

Mediante el OAM, puede controlar:

- El acceso a objetos de IBM MQ mediante la interfaz de cola de mensajes (MQI). Cuando un programa de aplicación intenta acceder a un objeto, el OAM comprueba que el perfil de usuario que realiza la solicitud posee autorización para la operación solicitada.

  En concreto, esto significa que las colas y los mensajes de las colas se pueden proteger contra accesos no autorizados.
- El permiso para utilizar mandatos PCF y MQSC.

Es posible que distintos grupos de usuarios tengan diferentes autorizaciones de acceso al mismo objeto. Por ejemplo, para una cola específica, un grupo podría realizar las operaciones de transferir y obtener; otro grupo podría tener autorización únicamente para examinar la cola (MQGET con la opción de examinar). De forma parecida, algunos grupos podrían tener autorización para transferir y obtener en una cola, pero no tenerla para modificar o suprimir la cola.

Mandatos IBM MQ for IBM i y realizar operaciones en objetos de IBM MQ for IBM i

IBM i Autorizaciones de IBM MQ en IBM i

Para acceder a objetos IBM MQ, necesita autorización para emitir el mandato y para acceder al objeto referenciado. Los administradores tienen acceso a todos los recursos de IBM MQ.

El acceso a los objetos de IBM MQ se controla mediante las autorizaciones para:

1. Emitir el mandato IBM MQ

2. Acceder a los objetos de IBM MQ a los que hace referencia el mandato

Todos los mandatos CL de IBM MQ for IBM i se suministran con un propietario de QMQM, y el perfil de administración (QMQMADM) tiene derechos *USE con el acceso *PUBLIC establecido en *EXCLUDE.

Nota: El programa instalador con licencia de IBM MQ para IBM i utiliza el programa QSRDUPER para duplicar objetos de mandato (*CMD) en QSYS. En IBM i V5R4 y posterior, el programa QSRDUPER se modificó de forma que el comportamiento predeterminado sea crear un mandato proxy en lugar de una duplicación de un mandato original. Un mandato proxy redirige la ejecución del mandato a otro mandato y tiene un atributo PRX. Si existe un mandato proxy en la biblioteca QSYS con el mismo nombre que el mandato que se está copiando, las autorizaciones privadas al mandato proxy no se otorgan al mandato en la biblioteca del producto. Los intentos de solicitar o ejecutar el mandato proxy en QSYS comprueban la autorización del mandato de destino en la biblioteca del producto. Todos los cambios en la autorización para objetos *CMD deben, por consiguiente, realizarse en la biblioteca del producto (QMQM) y los de QSYS no necesitan modificarse. Por ejemplo:

```
GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
```

Los cambios en la estructura de autorización de algunos de los mandatos CL del producto permiten el uso púbico de estos mandatos, si tiene la autorización del OAM necesaria sobre los objetos IBM MQ para realizar estos cambios.

Para ser un administrador de IBM MQ en IBM i, debe ser miembro del *grupo QMQMADM*. Este grupo tiene propiedades como las propiedades del grupo mqm en sistemas AIX, Linux, and Windows . En particular, el grupo QMQMADM se crea al instalar IBM MQ for IBM i y los miembros del grupo QMQMADM tienen acceso a todos los recursos de IBM MQ en el sistema. También tiene acceso a todos los recursos IBM MQ si tiene autorización *ALLOBJ.

Los administradores pueden utilizar mandatos CL para administrar IBM MQ. Uno de estos mandatos es GRTMQMAUT, que se utiliza para conceder autorizaciones a otros usuarios. Otro mandato, STRMQMMQSC, permite que un administrador emita mandatos MQSC a un gestor de colas local.

Conceptos relacionados

“Autorización para administrar IBM MQ en IBM i” en la página 95

IBM i *Autorizaciones de acceso para los objetos de IBM MQ en IBM i*

Autorizaciones de acceso necesarias para ejecutar mandatos CL de IBM MQ.

IBM MQ for IBM i categoriza los mandatos CL del producto en dos grupos:

Grupo 1

Los usuarios deben estar en el grupo de usuarios QMQMADM, o bien tener la autorización *ALLOBJ para procesar estos mandatos. Los usuarios que tienen una de estas autorizaciones pueden procesar todos los mandatos de todas las categorías y no necesitan ninguna otra autorización.

Nota: Estas autorizaciones alteran temporalmente cualquier autorización del OAM.

Estos mandatos se pueden agrupar como se indica a continuación:

- Mandatos de servidor de mandatos
  - ENDMQMCSVR, Finalizar el servidor de mandatos de IBM MQ
  - STRMQMCSVR, Iniciar el servidor de mandatos de IBM MQ
- Mandato de manejador de la cola de mensajes no entregados
  - STRMQMDLQ, Iniciar manejador de la cola de mensajes no entregados de IBM MQ
- Mandato de escucha
  - ENDMQMLSR, Finalizar escucha de IBM MQ
  - STRMQMLSR, Iniciar escucha no de objeto
- Mandatos de recuperación desde soporte

  - RCDMQMIMG, Registrar imagen de objeto de IBM MQ
  - RCRMQMOBJ, Volver a crear objeto de IBM MQ
  - WRKMQMTRN, Trabajar con transacciones de IBM MQ MQ
- Mandatos de gestor de colas
  - CRTMQM, Crear gestor de colas de mensajes
  - DLTMQM, Suprimir gestor de colas de mensajes
  - ENDMQM, Finalizar gestor de colas de mensajes
  - STRMQM, Iniciar gestor de colas de mensajes
- Mandatos de seguridad
  - GRTMQMAUT, Otorgar autorización sobre objeto de IBM MQ
  - RVKMQMAUT, Revocar autorización sobre objeto de IBM MQ
- Mandato de rastreo
  - TRCMQM, Rastrear trabajo de IBM MQ
- Mandatos de transacción
  - RSVMQMTRN, Resolver transacción de IBM MQ
- Mandatos de supervisor desencadenante
  - STRMQMTRM, Iniciar supervisor desencadenante
- Mandatos IBM MQSC
  - RUNMQSC, Ejecutar mandatos IBM MQSC
  - STRMQMMQSC, Iniciar mandatos IBM MQSC

Grupo 2

El resto de los mandatos, para los que se requiere dos niveles de autorización:

1. Autorización de IBM i para ejecutar el mandato. Un administrador de IBM MQ establece esto mediante el mandato **GRTOBJAUT** para alterar temporalmente la restricción *PUBLIC(*EXCLUDE) para un usuario o grupo de usuarios.

   Por ejemplo:

   ```
   GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
   ```

2. Autorización de IBM MQ para manipular los objetos de IBM MQ asociados con el mandato o los mandatos, dada la autorización correcta de IBM i en el paso 1.

   Esta autorización está controlada por el usuario que tiene la autorización adecuada del OAM para la acción necesaria, establecida por un administrador de IBM MQ mediante el mandato **GRTMQMAUT**

   Por ejemplo:

   ```
   GRTMQMAUT *connect authority to the queue manager + *admchg authority to
           the queue
   ```

Los mandatos se pueden agrupar como se indica a continuación:

- Mandatos de canal
  - CHGMQMCHL, Cambiar el canal de IBM MQ

    Requiere la autorización *connect sobre el gestor de colas y la autorización *admchg sobre el canal.
  - CPYMQMCHL, Copiar el canal de IBM MQ

Requiere la autorización *connect y *admcrt sobre el gestor de colas, la autorización *admdsp sobre el tipo de canal predeterminado que se va a copiar y la autorización *admcrt sobre la clase de objeto del canal.

Por ejemplo, para copiar un canal emisor, se necesita autorización *admdsp sobre el canal SYSTEM.DEF.SENDER

– CRTMQMCHL, Crear el canal de IBM MQ

Requiere la autorización *connect y *admcrt sobre el gestor de colas, la autorización *admdsp sobre el tipo de canal predeterminado que se va a crear y la autorización *admcrt sobre la clase de objeto del canal.

Por ejemplo, para crear un canal emisor, se necesita autorización *admdsp sobre el canal SYSTEM.DEF.SENDER

– DLTMQMCHL, Suprimir canal de IBM MQ

Requiere la autorización *connect sobre el gestor de colas y la autorización *admdlt sobre el canal.

– RSVMQMCHL, Resolver el canal de IBM MQ

Requiere la autorización *connect sobre el gestor de colas y la autorización *ctrlx sobre el canal.

- Mandatos de Visualizar

Para procesar los mandatos DSP, debe otorgar al usuario las autorizaciones `*connect` y `*admdsp` sobre el gestor de colas, junto con alguna de las opciones concretas que aparecen en la lista siguiente:

– DSPMQM, Visualizar gestor de colas de mensajes

– DSPMQMAUT, Visualizar autorización sobre objeto de IBM MQ

– DSPMQMAUTI, Visualizar información de autenticación de IBM MQ - `*admdsp` en el objeto de información de autenticación

– DSPMQMCHL, Visualizar el canal de IBM MQ - `*admdsp` sobre el canal

– DSPMQMCSVR, Visualizar el servidor de mandatos de IBM MQ

– DSPMQMNL, Visualizar lista de nombres de IBM MQ - `*admdsp` sobre la lista de nombres

– DSPMQMOBJN, Visualizar nombres de objeto de IBM MQ

– DSPMQMPRC, Visualizar proceso de IBM MQ - `*admdsp` sobre el proceso

– DSPMQMQ, Visualizar cola de IBM MQ - `*admdsp` sobre la cola

– DSPMQMTOP, Visualizar tema de IBM MQ - `*admdsp` sobre el tema

- Mandatos de Trabajar con

Para procesar los mandatos WRK y visualizar el panel de opciones, debe otorgar al usuario las autorizaciones `*connect` y `*admdsp` sobre el gestor de colas, junto con alguna de las opciones concretas que aparecen en la lista siguiente:

– WRKMQM, Trabajar con gestores de colas de mensajes

– WRKMQMAUT, Trabajar con autorización sobre objeto de IBM MQ

– WRKMQMAUTD, Trabajar con datos de autorización sobre objeto de IBM MQ

– WRKMQMAUTI, Trabajar con información de autenticación de IBM MQ

- `*admchg` para el mandato Cambiar objeto de información de autenticación de IBM MQ
- `*admcrt` para el mandato Crear y copiar objeto de información de autenticación de IBM MQ.
- `*admdlt` para el mandato Suprimir objeto de información de autenticación de IBM MQ.
- `*admdsp` mandato Visualizar objeto de información de autenticación de IBM MQ.

– WRKMQMCHL, Trabajar con canal de IBM MQ

Requiere las siguientes autorizaciones:

- *admchg para el mandato Cambiar canal de IBM MQ.
- *admclr para el mandato Borrar canal de IBM MQ.
- *admcrt para el mandato Crear y copiar canal de IBM MQ.
- *admdlt para el mandato Suprimir canal de IBM MQ.
- *admdsp para el mandato Visualizar canal de IBM MQ.
- *ctrl para el mandato Iniciar canal de IBM MQ.
- *ctrl para el mandato Finalizar canal de IBM MQ.
- *ctrl para el mandato Sondear canal de IBM MQ.
- *ctrlx para el mandato Restablecer canal de IBM MQ.
- *ctrlx para el mandato Resolver canal de IBM MQ.

– WRKMQMCHST, Trabajar con estado de canal de IBM MQ

  Requiere la autorización *admdsp sobre el canal.
– WRKMQMCL, Trabajar con clústeres de IBM MQ
– WRKMQMCLQ, Trabajar con colas de clúster de IBM MQ
– WRKMQMCLQM, Trabajar con el gestor de colas de clúster de IBM MQ
– WRKMQMLSR, Trabajar con escucha de IBM MQ
– WRKMQMMSG, Trabajar con mensajes de IBM MQ

  Requiere la autorización *browse sobre la cola
– WRKMQMNL, Trabajar con listas de nombres de IBM MQ

  Requiere las siguientes autorizaciones:
  - *admchg para el mandato Cambiar lista de nombres de IBM MQ.
  - *admcrt para el mandato Crear y copiar lista de nombres de IBM MQ.
  - *admdlt para el mandato Suprimir lista de nombres de IBM MQ.
  - *admdsp para el mandato Visualizar lista de nombres de IBM MQ.
– WRKMQMPRC, Trabajar con procesos de IBM MQ

  Requiere las siguientes autorizaciones:
  - *admchg para el mandato Cambiar proceso de IBM MQ.
  - *admcrt para el mandato Crear y copiar proceso de IBM MQ.
  - *admdlt para el mandato Suprimir proceso de IBM MQ.
  - *admdsp para el mandato Visualizar proceso de IBM MQ.
– WRKMQMQ, Trabajar con colas de IBM MQ

  Requiere las siguientes autorizaciones:
  - *admchg para el mandato Cambiar cola de IBM MQ.
  - *admclr para el mandato Borrar cola de IBM MQ.
  - *admcrt para el mandato Crear y copiar cola de IBM MQ.
  - *admdlt para el mandato Suprimir cola de IBM MQ.
  - *admdsp para el mandato Visualizar cola de IBM MQ.
– WRKMQMQSTS, Trabajar con estado de cola de IBM MQ
– WRKMQMTOP, Trabajar con temas de IBM MQ

  Requiere las siguientes autorizaciones
  - *admchg para el mandato Cambiar tema de IBM MQ.
  - *admcrt para el mandato Crear y copiar tema de IBM MQ.

- *admdlt para el mandato Suprimir tema de IBM MQ.
- *admdsp para el mandato Visualizar tema de IBM MQ.

– WRKMQMSUB, Trabajar con suscripciones de IBM MQ

- Otros mandatos de canal

  Para procesar los mandatos de canal, debe otorgar al usuario las autorizaciones específicas que aparecen en la lista siguiente:

  – ENDMQMCHL, Finalizar canal de IBM MQ

    Requiere la autorización *connect sobre el gestor de colas y la autorización *allmqi sobre la cola de transmisión asociada al canal.

  – ENDMQMLSR, Finalizar escucha de IBM MQ

    Esto requiere la autorización *connect sobre el gestor de colas y la autorización *ctrl sobre el objeto de escucha especificado.

  – PNGMQMCHL, Hacer ping en el canal de IBM MQ

    Esto requiere autorización *connect y *inq al gestor de objetos y autorización *ctrl al objeto de canal.

  – RSTMQMCHL, Restablecer canal de IBM MQ

    Requiere la autorización *connect sobre el gestor de colas.

  – STRMQMCHL, Iniciar canal de IBM MQ

    Esto requiere autorización *connect para el gestor de colas y autorización *ctrl para el objeto de canal.

  – STRMQMCHLI, Iniciar iniciador de canal de IBM MQ

    Requiere las autorizaciones *connect e *inq sobre el gestor de colas, y la autorización *allmqi sobre la cola de iniciación asociada a la cola de transmisión del canal.

  – STRMQMLSR, Iniciar escucha de IBM MQ

    Requiere la autorización *connect sobre el gestor de colas y la autorización *ctrl sobre el objeto de escucha especificado.

- Otros mandatos:

  Para procesar los mandatos siguientes, debe otorgar al usuario las autorizaciones específicas que aparecen en esta lista:

  – CCTMQM, Conectar a gestor de colas de mensajes

    Esto no necesita autoridad sobre objeto de IBM MQ.

  – CHGMQM, Cambiar gestor de colas de mensajes

    Esto requiere autorización *connect y *admchg para el gestor de colas.

  – CHGMQMAUTI, Cambiar información de autenticación de IBM MQ

    Requiere la autorización *connect sobre el gestor de colas y la autorización *admchg y *admdsp sobre el objeto de información de autenticación.

  – CHGMQMNL, Cambiar lista de nombres de IBM MQ

    Requiere la autorización *connect sobre el gestor de colas y *admchg sobre la lista de nombres.

  – CHGMQMPRC, Cambiar proceso de IBM MQ

    Requiere la autorización *connect sobre el gestor de colas y *admchg sobre el proceso.

  – CHGMQMQ, Cambiar cola de IBM MQ

    Requiere la autorización *connect sobre el gestor de colas y *admchg sobre la cola.

  – CLRMQMQ, Borrar cola de IBM MQ

Requiere la autorización *connect sobre el gestor de colas y *admclr sobre la cola.

- CPYMQMAUTI, Copiar información de autenticación de IBM MQ

  Requiere la autorización *connect sobre el gestor de colas y la autorización *admdsp sobre el objeto de información de autenticación y la autorización *admcrt sobre la clase de objeto de información de autenticación.

- CPYMQMNL, Copiar lista de nombres de IBM MQ

  Esto requiere autorización *connect y *admcrt para el gestor de colas.

- CPYMQMPRC, Copiar proceso de IBM MQ

  Esto requiere autorización *connect y *admcrt para el gestor de colas.

- CPYMQMQ, Copiar cola de IBM MQ

  Esto requiere autorización *connect y *admcrt para el gestor de colas.

- CRTMQMAUTI, Crear información de autenticación de IBM MQ

  Requiere la autorización *connect sobre el gestor de colas y la autorización *admdsp sobre el objeto de información de autenticación y la autorización *admcrt sobre la clase de objeto de información de autenticación.

- CRTMQMNL, Crear lista de nombres de IBM MQ

  Requiere la autorización *connect y *admcrt sobre el gestor de colas y la autorización *admdsp sobre la lista de nombres predeterminada.

- CRTMQMPRC, Crear proceso de IBM MQ

  Requiere la autorización *connect y *admcrt sobre el gestor de colas y *admdsp sobre el proceso predeterminado.

- CRTMQMQ, Crear cola de IBM MQ

  Requiere la autorización *connect y *admcrt sobre el gestor de colas y *admdsp sobre la cola predeterminada.

- CVTMQMDTA, mandato Convertir tipo de datos de IBM MQ

  Esto no necesita autoridad sobre objeto de IBM MQ.

- DLTMQMAUTI, Suprimir información de autenticación de IBM MQ

  Esto requiere autorización *connect para el gestor de colas y autorización *ctrlx para el objeto de información de autenticación.

- DLTMQMNL, Suprimir lista de nombres de IBM MQ

  Requiere la autorización *connect sobre el gestor de colas y *admdlt sobre la lista de nombres.

- DLTMQMPRC, Suprimir proceso de IBM MQ

  Requiere la autorización *connect sobre el gestor de colas y *admdlt sobre el proceso.

- DLTMQMQ, Suprimir cola de IBM MQ

  Requiere la autorización *connect sobre el gestor de colas y *admdlt sobre la cola.

- DSCMQM, Desconectar de gestor de colas de mensajes

  Esto no necesita autoridad sobre objeto de IBM MQ.

- RFRMQMAUT, Renovar seguridad

  Requiere la autorización *connect sobre el gestor de colas.

- RFRMQMCL, Renovar clúster

  Requiere la autorización *connect sobre el gestor de colas.

- RSMMQMCLQM, Reanudar gestor de colas de clúster

  Requiere la autorización *connect sobre el gestor de colas.

– RSTMQMCL, Restablecer clúster

  Requiere la autorización `*connect` sobre el gestor de colas.

– SPDMQMCLQM, Suspender gestor de colas de clúster

  Requiere la autorización `*connect` sobre el gestor de colas.

IBM i *Autorizaciones de acceso en IBM i*

Lea esta información para entender los mandatos de autorización de acceso.

Las autorizaciones definidas mediante la palabra clave AUT en los mandatos `GRTMQMAUT` y `RVKMQMAUT` se pueden clasificar de la siguiente manera:

- Autorizaciones relacionadas con las llamadas MQI
- Autorizaciones relacionadas con los mandatos de administración
- Autorizaciones de contexto
- Autorizaciones generales, es decir, para llamadas MQI, para mandatos o para ambos

Las tablas que figuran más abajo muestran las distintas autorizaciones que utilizan el parámetro AUT para llamadas MQI, llamadas de contexto, mandatos MQSC y PCF, y operaciones genéricas.

Tabla 15. Autorizaciones para llamadas MQI

AUT	Descripción
`*ALTUSR`	Permitir que se utilice la autorización de otro usuario para llamadas MQOPEN y MQPUT1.
`*BROWSE`	Recuperar un mensaje de una cola emitiendo una llamada MQGET con la opción BROWSE.
`*CONNECT`	Conectar la aplicación con el gestor de colas especificado emitiendo una llamada MQCONN.
`*GET`	Recuperar un mensaje de una cola emitiendo una llamada MQGET.
`*INQ`	Efectuar una consulta sobre una cola específica emitiendo una llamada MQINQ.
`*PUB`	Abrir un tema para publicar un mensaje utilizando una llamada MQPUT.
`*PUT`	Transferir un mensaje a una cola específica emitiendo una llamada MQPUT.
`*RESUME`	Reanudar una suscripción utilizando una llamada MQSUB.
`*SET`	Establecer los atributos de una cola de la MQI emitiendo una llamada MQSET. Si abre una cola para varias opciones, debe tener autorización sobre todas ellas.
`*SUB`	Crear, modificar o reanudar una suscripción en un tema utilizando una llamada MQSUB.

Tabla 16. Autorizaciones para llamadas de contexto

AUT	Descripción
`*PASSALL`	Pasar todo el contexto de la cola especificada. Todos los campos de contexto se copian de la solicitud original.
`*PASSID`	Pasar el contexto de identidad en la cola especificada. El contexto de identidad es el mismo que el de la solicitud.
`*SETALL`	Establecer todo el contexto de la cola especificada. Esta autorización la utilizan programas de utilidad especiales del sistema.

Tabla 16. Autorizaciones para llamadas de contexto (continuación)	
AUT	**Descripción**
*SETID	Establecer el contexto de identidad de la cola especificada. Esta autorización la utilizan programas de utilidad especiales del sistema.

Tabla 17. Autorizaciones para llamadas MQSC y PCF	
AUT	**Descripción**
*ADMCHG	Cambiar los atributos del objeto especificado.
*ADMCLR	Vaciar el objeto especificado (sólo el mandato PCF Vaciar objeto).
*ADMCRT	Crear objetos del tipo especificado.
*ADMDLT	Suprimir el objeto especificado.
*ADMDSP	Visualizar los atributos del objeto especificado.

Tabla 18. Autorizaciones para operaciones genéricas	
AUT	**Descripción**
*ALL	Utilizar todas las operaciones aplicables al objeto. La autorización `all` equivale a la unión de las autorizaciones `alladm`, `allmqi` y `system` adecuadas al tipo de objeto.
*ALLADM	Ejecutar todas las operaciones de administración aplicables al objeto.
*ALLMQI	Utilizar todas las llamadas MQI aplicables al objeto.
*CTRL	Controlar el arranque y el cierre de canales, escuchas y servicios.
*CTRLX	Restablecer el número de secuencia y resolver canales pendientes

IBM i *Utilización de los mandatos de autorización de acceso en IBM i*

Lea esta información para obtener información sobre los mandatos de autorización de acceso, y utilice los ejemplos de mandatos.

Utilización del mandato GRTMQMAUT

Si posee la autorización necesaria, puede utilizar el mandato GRTMQMAUT para otorgar a un perfil de usuario o grupo de usuarios autorización para acceder a un determinado objeto. Los ejemplos que figuran a continuación ilustran cómo se utiliza el mandato GRTMQMAUT:

1. ```
   GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
             AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
   ```

   En este ejemplo:

   - RED.LOCAL.QUEUE es el nombre del objeto.
   - *LCLQ (cola local) es el tipo de objeto.
   - GROUPA es el nombre de un perfil de usuario en el sistema para el que las autorizaciones se van a cambiar. Este perfil se puede utilizar como perfil de grupo para otros usuarios.
   - *BROWSE y *PUT son las autorizaciones que se van a otorgar sobre la cola especificada.

     *BROWSE añade autorización para examinar los mensajes de la cola (emitir MQGET con la opción de examinar).

     *PUT añade autorización para poner (MQPUT) mensajes en la cola.
   ```

- `saturn.queue.manager` es el nombre del gestor de colas.

2. El siguiente mandato otorga a los usuarios JACK y JILL todas las autorizaciones aplicables, sobre todas las definiciones de proceso, del gestor de colas predeterminado.

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
```

3. El siguiente mandato otorga al usuario GEORGE autorización para poner un mensaje en la cola ORDERS, en el gestor de colas TRENT.

```
GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
```

Utilización del mandato RVKMQMAUT

Si posee la autorización necesaria, puede utilizar el mandato RVKMQMAUT para eliminar del perfil de usuario o del grupo de usuarios una autorización ya otorgada para acceder a un determinado objeto. Los ejemplos que figuran a continuación ilustran cómo se utiliza el mandato RVKMQMAUT:

1.
```
RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
AUT(*PUT) MQMNAME('saturn.queue.manager')
```

   Se quita del grupo GROUPA la autorización (otorgada en el ejemplo anterior) para poner mensajes en la cola especificada.

2.
```
RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
MQMNAME(PAYROLLQM)
```

   La autorización para obtener mensajes procedentes de cualquier cola con un nombre que empiece por los caracteres PAY, que pertenece al gestor de colas PAYROLLQM, se elimina de todos los usuarios del sistema, a menos que ellos o un grupo al que pertenezcan, se hayan autorizado por separado.

Utilización del mandato DSPMQMAUT

El mandato Visualizar autorización de MQM (DSPMQMAUT) muestra, para el objeto y usuario especificados, la lista de autorizaciones que el usuario posee sobre el objeto. El siguiente ejemplo ilustra cómo se utiliza este mandato:

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

Utilización del mandato RFRMQMAUT

El mandato Renovar seguridad de MQM (RFRMQMAUT) le permite actualizar inmediatamente la información del grupo de autorizaciones del OAM, reflejando los cambios realizados a nivel de sistema operativo, sin necesidad de detener y reiniciar el gestor de colas. El siguiente ejemplo ilustra cómo se utiliza este mandato:

```
RFRMQMAUT MQMNAME(ADMINQM)
```

IBM i Tablas de especificación de autorizaciones en IBM i

Utilice esta información para determinar qué autorización es necesaria para utilizar llamadas API específicas, y opciones específicas de esas llamadas, en objetos de cola, objetos de proceso y objetos de gestor de colas.

Las tablas de especificación de autorizaciones en Tabla 19 en la página 175 definen de forma precisa cómo funcionan las autorizaciones y las restricciones que se aplican. Las tablas se aplican a estas situaciones:

- Aplicaciones que emiten llamadas MQI
- Programas de administración que emiten mandatos MQSC como mandatos PCF de escape
- Programas de administración que emiten mandatos PCF

En esta sección, la información se presenta como un conjunto de tablas que especifican los datos siguientes:

Acción que se va a realizar
Opción MQI, mandato MQSC o mandato PCF.

Objeto de control de acceso
Cola, definición de proceso, gestor de colas, lista de nombres, canal, canal de conexión de cliente, escucha, servicio u objeto de información de autenticación.

Autorización necesaria
Expresada como constante de tipo MQZAO_.

En las tablas, las constantes que tienen el prefijo MQZAO_ corresponden a las palabras clave en la lista de autorizaciones de los mandatos **GRTMQMAUT** y **RVKMQMAUT** de una determinada entidad. Por ejemplo, MQZAO_BROWSE corresponde con la palabra clave *BROWSE; asimismo, la palabra clave MQZAO_SET_ALL_CONTEXT corresponde con la palabra clave *SETALL y así sucesivamente. Estas constantes están definidas en el archivo de cabecera cmqzc.h que se proporciona con el producto.

Autorizaciones de MQI

Una aplicación puede emitir determinadas llamadas y opciones MQI sólo si el identificador de usuario bajo el que se está ejecutando (o cuyas autorizaciones puede asumir) tiene la autorización pertinente.

Hay cuatro llamadas MQI que pueden requerir comprobaciones de autorización: MQCONN, MQOPEN, MQPUT1 y MQCLOSE.

Para MQOPEN y MQPUT1, la comprobación de autorización se efectúa en el nombre del objeto que se está abriendo y no en el nombre o nombres resultantes de la resolución de nombre. Por ejemplo, una aplicación puede tener autorización para abrir una cola alias sin tener autorización para abrir la cola base en la que se resuelve la cola alias. La regla es que la comprobación se realiza en la primera definición encontrada durante el proceso de resolución de nombres que no es un alias de gestor de colas, a menos que la definición de alias de gestor de colas se abra directamente; es decir, su nombre aparece en el campo *ObjectName* del descriptor de objeto. La autoridad siempre es necesaria para el objeto concreto que se está abriendo; en algunos casos, es necesaria la autoridad adicional independiente de la cola obtenida a través de una autorización para el objeto del gestor de colas.

Tabla 19 en la página 175, Tabla 20 en la página 176, Tabla 21 en la página 176 y Tabla 22 en la página 177 resumen las autorizaciones necesarias para cada llamada.

Nota: En estas tablas no se mencionan listas de nombres, canales, canales de conexión de cliente, escuchas, servicios u objetos de información de autenticación. Esto se debe a que ninguna de las autorizaciones se aplica a estos objetos, salvo MQOO_INQUIRE, para la que se aplican las mismas autorizaciones que para los demás objetos.

Tabla 19. Autorización de seguridad necesaria para llamadas MQCONN

Autorización necesaria para:	Objeto de cola (“1” en la página 177)	Objeto de proceso	Objeto gestor de colas
Opción MQCONN	No aplicable	No aplicable	MQZAO_CONNECT

Tabla 20. Autorización de seguridad necesaria para llamadas MQOPEN			
Autorización necesaria para:	**Objeto de cola (“1” en la página 177)**	**Objeto de proceso**	**Objeto gestor de colas**
MQOO_INQUIRE	MQZAO_INQUIRE **(“2” en la página 177)**	MQZAO_INQUIRE **(“2” en la página 177)**	MQZAO_INQUIRE **(“2” en la página 177)**
MQOO_BROWSE	MQZAO_BROWSE	No aplicable	No se comprueba
MQOO_INPUT_*	MQZAO_INPUT	No aplicable	No se comprueba
MQOO_SAVE_ ALL_CONTEXT **(“3” en la página 177)**	MQZAO_INPUT	No aplicable	No aplicable
MQOO_OUTPUT (Cola normal) **(“4” en la página 177)**	MQZAO_OUTPUT	No aplicable	No aplicable
MQOO_PASS_ IDENTITY_CONTEXT **(“5” en la página 177)**	MQZAO_PASS_ IDENTITY_CONTEXT	No aplicable	No se comprueba
MQOO_PASS_ALL_ CONTEXT **(“5” en la página 177, “6” en la página 177)**	MQZAO_PASS _ALL_CONTEXT	No aplicable	No se comprueba
MQOO_SET_ IDENTITY_CONTEXT **(“5” en la página 177, “6” en la página 177)**	MQZAO_SET_ IDENTITY_CONTEXT	No aplicable	MQZAO_SET_ IDENTITY_CONTEXT **(“7” en la página 177)**
MQOO_SET_ ALL_CONTEXT **(“5” en la página 177, “8” en la página 177)**	MQZAO_SET_ ALL_CONTEXT	No aplicable	MQZAO_SET_ ALL_CONTEXT **(“7” en la página 177)**
MQOO_OUTPUT (cola de transmisión) **(“9” en la página 177)**	MQZAO_SET_ ALL_CONTEXT	No aplicable	MQZAO_SET_ ALL_CONTEXT **(“7” en la página 177)**
MQOO_SET	MQZAO_SET	No aplicable	No se comprueba
MQOO_ALTERNATE_ USER_AUTHORITY	**(“10” en la página 178)**	**(“10” en la página 178)**	MQZAO_ALTERNATE_ USER_AUTHORITY **(“10” en la página 178, “11” en la página 178)**

Tabla 21. Autorización de seguridad necesaria para llamadas MQPUT1			
Autorización necesaria para:	**Objeto de cola (“1” en la página 177)**	**Objeto de proceso**	**Objeto gestor de colas**
MQPMO_PASS_ IDENTITY_CONTEXT	MQZAO_PASS_ IDENTITY_CONTEXT **(“12” en la página 178)**	No aplicable	No se comprueba
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **(“12” en la página 178)**	No aplicable	No se comprueba

Tabla 21. Autorización de seguridad necesaria para llamadas MQPUT1 (continuación)			
Autorización necesaria para:	**Objeto de cola (“1” en la página 177)**	**Objeto de proceso**	**Objeto gestor de colas**
MQPMO_SET_ IDENTITY_CONTEXT	MQZAO_SET_ IDENTITY_CONTEXT **(“12” en la página 178)**	No aplicable	MQZAO_SET_ IDENTITY_CONTEXT **(“7” en la página 177)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **(“12” en la página 178)**	No aplicable	MQZAO_SET_ ALL_CONTEXT **(“7” en la página 177)**
(Cola de transmisión) **(“9” en la página 177)**	MQZAO_SET_ ALL_CONTEXT	No aplicable	MQZAO_SET_ ALL_CONTEXT **(“7” en la página 177)**
MQPMO_ALTERNATE_ USER_AUTHORITY	**(“13” en la página 178)**	No aplicable	MQZAO_ALTERNATE_ USER_AUTHORITY **(“11” en la página 178)**

Tabla 22. Autorización de seguridad necesaria para llamadas MQCLOSE			
Autorización necesaria para:	**Objeto de cola (“1” en la página 177)**	**Objeto de proceso**	**Objeto gestor de colas**
MQCO_DELETE	MQZAO_DELETE **(“14” en la página 178)**	No aplicable	No aplicable
MQCO_DELETE _PURGE	MQZAO_DELETE **(“14” en la página 178)**	No aplicable	No aplicable

Notas para las tablas:

1. Si se va abrir una cola modelo:
   - Para la cola modelo, es necesaria la autorización MQZAO_DISPLAY además de la autorización para abrir la cola modelo correspondiente al tipo de acceso para el que se está efectuando la apertura.
   - La autorización MQZAO_CREATE no es necesaria para crear la cola dinámica.
   - El identificador de usuario utilizado para abrir la cola modelo se otorga automáticamente a todas las autorizaciones específicas de la cola (equivalentes a MQZAO_ALL) para la cola dinámica creada.
2. Se comprueba el objeto de cola, proceso, lista de nombres o gestor de colas, dependiendo del tipo de objeto que vaya a abrirse.
3. También debe especificarse MQOO_INPUT_*. Esta opción es válida para una cola local, modelo o alias.
4. Esta comprobación se realiza en todos los casos de salida, excepto en el caso especificado en la nota “9” en la página 177.
5. También debe especificarse MQOO_OUTPUT.
6. Esta opción también implica MQOO_PASS_IDENTITY_CONTEXT.
7. Esta autorización es necesaria tanto para el objeto gestor de colas como para la cola concreta.
8. Esta opción también implica MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT y MQOO_SET_IDENTITY_CONTEXT.
9. Esta comprobación se realiza para una cola local o modelo cuyo atributo de cola *Usage* sea MQUS_TRANSMISSION y se esté abriendo directamente para salida. Esto no es aplicable si se abre una cola remota (especificando los nombres del gestor de colas remoto y la cola remota, o especificando el nombre de una definición local de la cola remota).

10. También debe especificarse como mínimo una de las opciones MQOO_INQUIRE (para cualquier tipo de objeto) o (para colas) MQOO_BROWSE, MQOO_INPUT_*, MQOO_OUTPUT o MQOO_SET. La comprobación que se lleva a cabo es la misma que en las otras opciones especificadas, utilizando el identificador de usuario alternativo suministrado para la autorización sobre el objeto específico nombrado, y la autorización sobre la aplicación actual para la comprobación MQZAO_ALTERNATE_USER_IDENTIFIER.
11. Esta autorización permite especificar cualquier *AlternateUserId*.
12. También se realiza una comprobación MQZAO_OUTPUT si la cola no tiene un atributo de cola *Usage* de MQUS_TRANSMISSION.
13. La comprobación que se lleva a cabo es la misma que en las otras opciones especificadas, utilizando el identificador de usuario alternativo suministrado para la autorización sobre la cola nombrada, y la autorización sobre la aplicación actual para la comprobación MQZAO_ALTERNATE_USER_IDENTIFIER.
14. La comprobación solo se lleva a cabo si se cumplen las dos sentencias siguientes:
    - Se está cerrando y suprimiendo una cola dinámica permanente.
    - La cola no ha sido creado por la llamada a MQOPEN que ha devuelto el descriptor de contexto de objeto que se utiliza.

    De lo contrario, no hay comprobación.

Notas generales:

1. La autorización especial MQZAO_ALL_MQI incluye todas las autorizaciones siguientes que sean aplicables al 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 (vea la nota “14” en la página 178) y MQZAO_DISPLAY están clasificadas como autorizaciones de administración. Por lo tanto no están incluidas en MQZAO_ALL_MQI.
3. *No se comprueba* significa que no se lleva a cabo la comprobación.
4. *No es aplicable* significa que la comprobación de autorización no tiene sentido en esta operación. Por ejemplo, no se puede emitir una llamada MQPUT dirigida a un objeto proceso.

IBM i *Autorizaciones para mandatos MQSC en los PCF de escape en IBM i*

Estas autorizaciones permiten a un usuario emitir mandatos de administración como un mensaje PCF de escape. Estos métodos permiten a un programa enviar un mandato de administración como un mensaje a un gestor de colas, para que se ejecute en nombre de dicho usuario.

Esta sección resume las autorizaciones necesarias para cada mandato MQSC contenido en un PCF de escape.

No es aplicable significa que la comprobación de autorización no tiene sentido en esta operación.

El ID de usuario bajo el que se ejecuta el programa que envía el mandato también debe tener las autorizaciones siguientes:

- Autorización MQZAO_CONNECT para el gestor de colas
- Autorización DISPLAY sobre el gestor de colas para realizar mandatos PCF
- Autorización para emitir mandatos MQSC dentro del texto del mandato PCF de escape

ALTER *objeto*

Objeto	**Autorización necesaria**
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	MQZAO_CHANGE
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE

CLEAR *objeto*

Objeto	**Autorización necesaria**
Cola	MQZAO_CLEAR
Tema	MQZAO_CLEAR
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	No aplicable
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

DEFINE *objeto* NOREPLACE (“1” en la página 182)

Objeto	**Autorización necesaria**
Cola	MQZAO_CREATE **(“2” en la página 183)**
Tema	MQZAO_CREATE **(“2” en la página 183)**
Proceso	MQZAO_CREATE **(“2” en la página 183)**
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CREATE **(“2” en la página 183)**
Información de autenticación	MQZAO_CREATE **(“2” en la página 183)**
Canal	MQZAO_CREATE **(“2” en la página 183)**

Objeto	Autorización necesaria
Canal de conexión de cliente	MQZAO_CREATE **(“2” en la página 183)**
Escucha	MQZAO_CREATE **(“2” en la página 183)**
Servicio	MQZAO_CREATE **(“2” en la página 183)**

DEFINE *objeto* REPLACE (“1” en la página 182, “3” en la página 183)

Objeto	Autorización necesaria
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE

DELETE *objeto*

Objeto	Autorización necesaria
Cola	MQZAO_DELETE
Tema	MQZAO_DELETE
Proceso	MQZAO_DELETE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_DELETE
Información de autenticación	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de conexión de cliente	MQZAO_DELETE
Escucha	MQZAO_DELETE
Servicio	MQZAO_DELETE

DISPLAY *objeto*

Objeto	Autorización necesaria
Cola	MQZAO_DISPLAY
Tema	MQZAO_DISPLAY
Proceso	MQZAO_DISPLAY
Gestor de colas	MQZAO_DISPLAY
Lista de nombres	MQZAO_DISPLAY

Objeto	Autorización necesaria
Información de autenticación	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de conexión de cliente	MQZAO_DISPLAY
Escucha	
Servicio	

PING CHANNEL

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

RESET CHANNEL

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL_EXTENDED
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

RESOLVE CHANNEL

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable

Objeto	Autorización necesaria
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL_EXTENDED
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

START *objeto*

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	MQZAO_CONTROL
Servicio	MQZAO_CONTROL

STOP *objeto*

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	MQZAO_CONTROL
Servicio	MQZAO_CONTROL

Nota:

1. Para los mandatos DEFINE, se necesita también la autorización MQZAO_DISPLAY sobre el objeto LIKE, si se ha especificado uno, o sobre el objeto SYSTEM.DEFAULT.xxx adecuado si se ha omitido LIKE.

2. La autorización MQZAO_CREATE no es específica de un objeto o tipo de objeto en particular. La autorización para crear se otorga sobre todos los objetos de un gestor de colas indicado, especificando el tipo de objeto QMGR en el mandato GRTMQMAUT.
3. Esta opción es aplicable si el objeto que va a sustituirse ya existe. Si no existe, la comprobación es como para DEFINE *objeto* NOREPLACE.

IBM i *Autorizaciones para mandatos PCF en IBM i*

Estas autorizaciones permiten a un usuario emitir mandatos de administración como mandatos PCF. Estos métodos permiten a un programa enviar un mandato de administración como un mensaje a un gestor de colas, para que se ejecute en nombre de dicho usuario.

Esta sección resume las autorizaciones necesarias para cada mandato PCF.

La indicación *No se comprueba* significa que no se lleva a cabo ninguna comprobación de autorización; *No aplicable* significa que la comprobación de autorización no es pertinente en esta operación.

El ID de usuario bajo el que se ejecuta el programa que envía el mandato también debe tener las autorizaciones siguientes:

- Autorización MQZAO_CONNECT para el gestor de colas
- Autorización DISPLAY sobre el gestor de colas para realizar mandatos PCF

La autorización especial MQZAO_ALL_ADMIN incluye las siguientes autorizaciones:

- MQZAO_CHANGE
- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY
- MQZAO_CONTROL
- MQZAO_CONTROL_EXTENDED

MQZAO_CREATE no está incluida porque no es específica de un objeto o tipo de objeto en particular.

Change *objeto*

Objeto	**Autorización necesaria**
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	MQZAO_CHANGE
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE

Clear *objeto*

Objeto	**Autorización necesaria**
Cola	MQZAO_CLEAR
Tema	MQZAO_CLEAR

Objeto	Autorización necesaria
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	No aplicable
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

Copy *objeto* (without replace) ("1" en la página 188)

Objeto	Autorización necesaria
Cola	MQZAO_CREATE **("2" en la página 188)**
Tema	MQZAO_CREATE **("2" en la página 188)**
Proceso	MQZAO_CREATE **("2" en la página 188)**
Gestor de colas	No aplicable
NamelistMQZAO_CREATE	MQZAO_CREATE **("2" en la página 188)**
Información de autenticación	MQZAO_CREATE **("2" en la página 188)**
Canal	MQZAO_CREATE **("2" en la página 188)**
Canal de conexión de cliente	MQZAO_CREATE **("2" en la página 188)**
Escucha	MQZAO_CREATE **("2" en la página 188)**
Servicio	MQZAO_CREATE **("2" en la página 188)**

Copy *objeto* (with replace) ("1" en la página 188, "4" en la página 189)

Objeto	Autorización necesaria
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE

Create *objeto* (without replace) (“3” en la página 188)

Objeto	Autorización necesaria
Cola	MQZAO_CREATE **(“2” en la página 188)**
Tema	MQZAO_CREATE **(“2” en la página 188)**
Proceso	MQZAO_CREATE **(“2” en la página 188)**
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CREATE **(“2” en la página 188)**
Información de autenticación	MQZAO_CREATE **(“2” en la página 188)**
Canal	MQZAO_CREATE **(“2” en la página 188)**
Canal de conexión de cliente	MQZAO_CREATE **(“2” en la página 188)**
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE

Create *objeto* (with replace) (“3” en la página 188, “4” en la página 189)

Objeto	Autorización necesaria
Cola	MQZAO_CHANGE
Tema	MQZAO_CHANGE
Proceso	MQZAO_CHANGE
Gestor de colas	No aplicable
Lista de nombres	MQZAO_CHANGE
Información de autenticación	MQZAO_CHANGE
Canal	MQZAO_CHANGE
Canal de conexión de cliente	MQZAO_CHANGE
Escucha	MQZAO_CHANGE
Servicio	MQZAO_CHANGE

Delete *objeto*

Objeto	Autorización necesaria
Cola	MQZAO_DELETE
Tema	MQZAO_DELETE
Proceso	MQZAO_DELETE
Gestor de colas	MQZAO_DELETE
Lista de nombres	MQZAO_DELETE
Información de autenticación	MQZAO_DELETE
Canal	MQZAO_DELETE
Canal de conexión de cliente	MQZAO_DELETE
Escucha	MQZAO_DELETE

Objeto	Autorización necesaria
Servicio	MQZAO_DELETE

Inquire *objeto*

Objeto	Autorización necesaria
Cola	MQZAO_DISPLAY
Tema	MQZAO_DISPLAY
Proceso	MQZAO_DISPLAY
Gestor de colas	MQZAO_DISPLAY
Lista de nombres	MQZAO_DISPLAY
Información de autenticación	MQZAO_DISPLAY
Canal	MQZAO_DISPLAY
Canal de conexión de cliente	MQZAO_DISPLAY
Escucha	MQZAO_DISPLAY
Servicio	MQZAO_DISPLAY

Inquire *objeto* names

Objeto	Autorización necesaria
Cola	No se comprueba
Tema	No se comprueba
Proceso	No se comprueba
Gestor de colas	No se comprueba
Lista de nombres	No se comprueba
Información de autenticación	No se comprueba
Canal	No se comprueba
Canal de conexión de cliente	No se comprueba
Escucha	No se comprueba
Servicio	No se comprueba

Sondear canal

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL

Objeto	Autorización necesaria
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

Restablecer canal

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL_EXTENDED
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

Restablecer estadísticas de la cola

Objeto	Autorización necesaria
Cola	MQZAO_DISPLAY y MQZAO_CHANGE
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	No aplicable
Canal de conexión de cliente	No aplicable
Escucha	
Servicio	

Resolver canal

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable

Objeto	Autorización necesaria
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL_EXTENDED
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

Iniciar canal

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

Detener canal

Objeto	Autorización necesaria
Cola	No aplicable
Tema	No aplicable
Proceso	No aplicable
Gestor de colas	No aplicable
Lista de nombres	No aplicable
Información de autenticación	No aplicable
Canal	MQZAO_CONTROL
Canal de conexión de cliente	No aplicable
Escucha	No aplicable
Servicio	No aplicable

Nota:

1. En los mandatos Copy, también es necesaria la autorización MQZAO_DISPLAY para el objeto de origen.
2. La autorización MQZAO_CREATE no es específica de un objeto o tipo de objeto en particular. La autorización para crear se otorga sobre todos los objetos de un gestor de colas indicado, especificando el tipo de objeto QMGR en el mandato GRTMQMAUT.
3. Para los mandatos Create, también se necesita la autorización MQZAO_DISPLAY para el SYSTEM.DEFAULT.* .

4. Esta opción es aplicable si el objeto que va a sustituirse ya existe. Si no existe, la comprobación es como para un mandato Copy o Create sin sustitución.

IBM i Perfiles OAM genéricos en IBM i

Los perfiles genéricos del gestor de autorizaciones sobre objetos (OAM) le permiten establecer de una sola vez la autorización que un usuario tiene sobre muchos objetos, en lugar de tener que emitir mandatos **GRTMQMAUT** distintos para cada objeto individual en el momento de su creación. La utilización de perfiles genéricos en el mandato **GRTMQMAUT** permite establecer una autorización genérica para todos los objetos que se creen en el futuro que se ajusten a dicho perfil.

El resto de esta sección describe con más detalle el uso de los perfiles genéricos:

- "Utilización de caracteres comodín" en la página 189
- "Prioridades de perfiles" en la página 189

Utilización de caracteres comodín

Lo que hace que un perfil sea genérico es el uso de caracteres especiales (caracteres comodín) en el nombre del perfil. Por ejemplo, el carácter comodín de signo de interrogación (?) coincide con cualquier carácter individual de un nombre. Por tanto, si se especifica `ABC.?EF`, la autorización que se otorga a dicho perfil se aplica a todos los objetos creados con los nombres `ABC.DEF`, `ABC.CEF`, `ABC.BEF`, etcétera.

Los caracteres comodín disponibles son:

?

Utilice el signo de interrogación (?) en lugar de cualquier otro carácter. Por ejemplo, `AB.?D` se aplicaría a los objetos `AB.CD`, `AB.ED`y `AB.FD`.

*****

Utilice el asterisco (*) como:

- Un *calificador* de un nombre de perfil para que coincida con cualquier calificador de un nombre de objeto. Un calificador es la parte de un nombre de objeto delimitada por un punto. Por ejemplo, en `ABC.DEF.GHI`, los calificadores son `ABC`, `DEF` y `GHI`.

  Por ejemplo, `ABC.*.JKL` se aplicaría a los objetos `ABC.DEF.JKL`y `ABC.GHI.JKL`. Tenga en cuenta que **no** se aplicará `ABC.JKL`. Cuando se utiliza el carácter * en este contexto siempre indica un calificador.
- Un carácter contenido en un calificador de un nombre de perfil para que coincida con cero o más caracteres incluidos en el calificador de un nombre de objeto.

  Por ejemplo, `ABC.DE*.JKL` se aplicaría a los objetos `ABC.DE.JKL`, `ABC.DEF.JKL`y `ABC.DEGH.JKL`.

******

Utilice el asterisco doble (**) ***una vez*** en el nombre de un perfil como:

- El nombre de perfil completo para que coincida con todos los nombres de objetos. Por ejemplo, si utiliza la palabra clave `OBJTYPE (*PRC)` para identificar procesos y luego utiliza ** como el nombre del perfil, se cambian las autorizaciones de todos los procesos.
- Como cualquier calificador del principio, mitad o final de un nombre de perfil para que coincida con cero o más calificadores contenidos en un nombre de objeto. Por ejemplo, `**.ABC` identifica todos los objetos con el calificador final ABC.

Prioridades de perfiles

Una cuestión importante que debe comprender cuando utilice perfiles genéricos es la prioridad que se otorga a los perfiles a la hora de decidir qué autorizaciones se han de aplicar a un objeto que se está creando. Por ejemplo, suponga que ha emitido los mandatos:

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

La primera otorga autorización de colocación a todas las colas para el principal FRED con nombres que coinciden con el perfil AB. *; el segundo otorga autorización de obtención sobre los mismos tipos de cola que coinciden con el perfil AB.C*.

Suponga que ahora crea una cola con el nombre AB.CD. De acuerdo con las reglas para las comparaciones con comodines, cualquiera de los dos mandatos GRTMQMAUT podría aplicarse a dicha cola. Por lo tanto, la cuestión es ¿tiene autorización para transferir o para obtener?

Para encontrar la respuesta, aplique la regla según la cual cuando varios perfiles pueden aplicarse a un objeto, **sólo se aplica el más específico**. El modo en que se aplica esta regla es comparando los nombres de perfil de izquierda a derecha. Siempre que difieren, un carácter no genérico es más especifico que un carácter genérico. De este modo, en el ejemplo anterior, la cola AB.CD tiene autorización para **obtener** (AB.C* es más específico que AB.*).

Cuando se comparan caracteres genéricos, el orden de *especificidad* es el siguiente:

1. ?
2. *
3. **

IBM i Especificación del servicio de autorización instalado en IBM i

Puede especificar el componente de servicio de autorización que se ha de utilizar.

El parámetro **Service Component name** en **GRTMQMAUT** y **RVKMQMAUT** le permite especificar el nombre del componente de servicio de autorización instalado.

Seleccionar **F24** en el panel inicial, seguido de **F9=Todos los parámetros** en el siguiente panel de cualquiera de los dos mandatos, permite especificar el componente de autorización instalado (*DFT) o bien el nombre del componente de servicio de autorización necesario especificado en la sección Service del archivo qm.ini del gestor de colas.

DSPMQMAUT también tiene este parámetro adicional. Este parámetro permite buscar el nombre de objeto, el tipo de objeto y el usuario especificados en todos los componentes de autorización instalados (*DFT), o bien en el nombre del componente de servicio de autorización especificado.

IBM i Trabajar con y sin perfiles de autorización en IBM i

Utilice esta información para aprender a trabajar con perfiles de autorización y a trabajar sin perfiles de autorización.

Puede trabajar con perfiles de autorización, como se explica en “Trabajar con perfiles de autorización” en la página 190, o sin ellos, como se explica a continuación:

Para trabajar sin perfiles de autorización, utilice *NONE como parámetro Authority en **GRTMQMAUT** para crear perfiles sin autorización. Los perfiles existentes se quedan igual.

En **RVKMQMAUT**, utilice *REMOVE como parámetro Authority para eliminar un perfil de autorización existente.

Trabajar con perfiles de autorización

Hay dos mandatos asociados con la creación de perfiles de autorización:

- **WRKMQMAUT**
- **WRKMQMAUTD**

Puede acceder a estos mandatos directamente desde la línea de mandatos o bien desde el panel WRKMQM realizando lo siguiente:

1. Escriba el nombre del gestor de colas y pulse la tecla `Enter` para acceder al panel de resultados de **WRKMQM** .
2. Seleccionando `F23=More options` en este panel.

La opción 24 selecciona el panel de resultados para el mandato **WRKMQMAUT** y la opción 25 selecciona el mandato **WRKMQMAUTI** , que se utiliza con la capa de enlaces SSL.

WRKMQMAUT

Este mandato permite trabajar con los datos de autorizaciones que se guardan en la cola de autorizaciones.

Nota: Para ejecutar este mandato, debe tener autorización `*connect` y `*admdsp` sobre el gestor de colas. Sin embargo, para crear o suprimir un perfil, necesita la autorización QMQMADM.

Si envía la información a la pantalla, se visualiza una lista de nombres de perfiles de autorización junto con sus tipos. Si imprime la salida, recibirá una lista detallada de todos los datos de autorizaciones, los usuarios registrados y sus autorizaciones.

Al especificar un nombre de objeto o perfil en este panel, y al pulsar INTRO se le lleva al panel de resultados de **WRKMQMAUT** .

Si selecciona `4=Delete`, vaya a un nuevo panel desde el que puede confirmar que desea suprimir todos los nombres de usuario registrados en el nombre de perfil de autorización genérico que especifique. Esta opción ejecuta el mandato **RVKMQMAUT** con la opción *REMOVE para todos los usuarios y **sólo** se aplica a los nombres de perfiles genéricos.

Si selecciona `12=Work with profile` , vaya al panel de resultados del mandato **WRKMQMAUTD** , tal como se explica en “WRKMQMAUTD” en la página 191.

WRKMQMAUTD

Este mandato permite visualizar todos los usuarios registrados con un nombre de perfil de autorización y un tipo de objeto determinados. Para ejecutar este mandato, debe tener autorización `*connect` y `*admdsp` sobre el gestor de colas. Sin embargo, para otorgar, ejecutar, crear o suprimir un perfil, necesita la autorización QMQMADM.

Al seleccionar `F24=More keys` en el panel de entrada inicial, seguido de la opción `F9=All Parameters` se muestra el nombre de componente de servicio como para **GRTMQMAUT** y **RVKMQMAUT**.

Nota: La clave `F11=Display Object Authorizations` conmuta entre los siguientes tipos de autorizaciones:

- Autorizaciones de objetos
- Autorizaciones de contexto
- Autorizaciones de MQI

Las opciones que aparecen en la pantalla son:

2=Grant
Le lleva al panel del mandato **GRTMQMAUT** para añadirlo a las autorizaciones actuales.

3=Revoke
Le lleva al panel **RVKMQMAUT** para eliminar algunas de las definiciones actuales.

4=Delete
Le lleva a un panel que le permite suprimir los datos de autorizaciones de usuarios especificados. Ejecuta el mandato **RVKMQMAUT** con la opción *REMOVE.

5=Display
Le lleva al mandato **DSPMQMAUT** existente.

F6=Create
Le lleva al panel de **GRTMQMAUT** que le permite crear un registro de autorización de perfil.

IBM i Directrices para el Gestor de autorizaciones sobre objetos (OAM) en IBM i

Consejos y sugerencias adicionales para utilizar el gestor de autorizaciones sobre objetos (OAM)

Limitar el acceso a operaciones confidenciales

Algunas operaciones son confidenciales; limítelas a los usuarios con privilegios. Por ejemplo,

- Acceder a algunas colas especiales, tales como colas de transmisión o la cola de mandatos `SYSTEM.ADMIN.COMMAND.QUEUE`
- La ejecución de programas que utilicen todas las opciones de contexto de la MQI
- Crear y copiar colas de aplicación

Directorios del gestor de colas

Los directorios y las bibliotecas que contiene las colas y otros datos de los gestores de colas son privados del producto. No utilice mandatos del sistema operativo estándar para otorgar o revocar autorizaciones sobre recursos de la MQI.

Colas

La autorización sobre una cola dinámica se basa en la cola modelo de la que se deriva, aunque no tiene por qué ser igual.

En las colas alias y las colas remotas, la autorización es la del objeto propiamente dicho, no la de la cola en la que se resuelve la cola alias o remota. Es posible otorgar a un perfil de usuario autorización para que acceda a una cola alias que se resuelve en una cola local sobre la que el perfil de usuario no tenga permiso de acceso.

Limite la autorización para crear colas a los usuarios con privilegios. Si no lo hace así, los usuarios pueden eludir el control de acceso normal creando un alias.

Autorización de usuario alternativo

La autorización de usuario alternativo controla si un perfil de usuario puede utilizar la autorización de otro perfil de usuario al acceder a un objeto IBM MQ. Esta técnica es esencial cuando un servidor recibe solicitudes de un programa y el servidor desea asegurarse que el programa tiene la autorización necesaria para la solicitud. El servidor puede tener la autorización necesaria, pero necesita saber si el programa tiene autorización para las acciones que ha solicitado.

Por ejemplo:

- Un programa servidor que se está ejecutando bajo el perfil PAYSERV recupera de una cola un mensaje de solicitud que el perfil de usuario USER1 puso en la cola.
- Cuando el programa servidor obtiene el mensaje de solicitud, procesa la solicitud y vuelve a transferir la respuesta a la cola de respuestas especificada con el mensaje de solicitud.
- En lugar de utilizar su propio perfil de usuario (PAYSERV) para autorizar la apertura de una cola de respuestas, el servidor puede especificar algún otro perfil de usuario, en este caso, USER1. En este ejemplo, se puede emplear la autorización de usuario alternativo para controlar si PAYSERV tiene autorización para especificar USER1 como perfil de usuario alternativo al abrir la cola de respuestas.

El perfil de usuario alternativo se especifica en el campo *`AlternateUserId`* del descriptor de objeto.

Nota: Puede utilizar perfiles de usuario alternativo en cualquier objeto IBM MQ. El uso de un perfil de usuario alternativo no afecta al perfil de usuario utilizado por cualquier otro gestor de recursos.

Autorización de contexto

El contexto es la información que se aplica a un mensaje determinado y está contenida en el descriptor de mensaje, MQMD, que forma parte del mensaje.

Para obtener descripciones de los campos de descriptor de mensaje relacionados con el contexto, consulte MQMD-Descriptor de mensaje.

Para obtener información sobre las opciones de contexto, consulte Contexto de mensaje.

Consideraciones sobre la seguridad remota

Para la seguridad remota, tenga en cuenta lo siguiente:

Autoridad de transferencia
Por razones de seguridad entre los gestores de colas, puede especificar la autorización para transferir que se utiliza cuando un canal recibe un mensaje enviado desde otro gestor de colas.

Este parámetro sólo es válido para tipos de canal RCVR, RQSTR o CLUSRCVR. Especifique el atributo de canal PUTAUT como se indica a continuación:

DEF
Perfil de usuario predeterminado. Es el perfil de usuario QMQM bajo el que se está ejecutando el agente de canal de mensajes.

CTX
El perfil de usuario del contexto de mensaje.

Colas de transmisión
Los gestores de colas transfieren automáticamente los mensajes remotos a una cola de transmisión; no se requiere ninguna autorización especial. Sin embargo, se necesita una autorización especial para transferir un mensaje directamente a una cola de transmisión.

Salidas de canal
Las salidas de canal se pueden utilizar para implementar medidas de seguridad adicionales.

Registros de autenticación de canal
Se utiliza para ejercer un control más preciso sobre el acceso otorgado a la conexión de sistemas a nivel de canal.

Si desea más información sobre la seguridad remota, consulte “Autorización de canal” en la página 120.

Protección de canales con SSL/TLS

El protocolo TLS (seguridad de la capa de transporte) proporciona seguridad de canal, con protección contra escuchas y manipulaciones no autorizadas y contra falsas identidades. El soporte de IBM MQ para TLS le permite especificar, en la definición de canal, que un canal determinado utilice seguridad TLS. También puede especificar detalles de la seguridad que desea, como por ejemplo el algoritmo de cifrado que desea utilizar.

El soporte de TLS en IBM MQ utiliza el *objeto de información de autenticación* del gestor de colas y diversos mandatos CL y MQSC, así como parámetros de gestor de colas y canal que definen detalladamente el soporte de TLS necesario.

Los siguientes mandatos CL tienen soporte para TLS:

WRKMQMAUTI
Trabajar con los atributos de un objeto de información de autenticación.

CHGMQMAUTI
Modificar los atributos de un objeto de información de autenticación.

CRTMQMAUTI
Crear un objeto de información de autenticación.

CPYMQMAUTI
Crear un objeto de información de autenticación copiando uno existente.

DLTMQMAUTI
Suprimir un objeto de información de autenticación.

DSPMQMAUTI
Visualiza los atributos de un objeto de información de autenticación específico.

Para obtener una visión general de la seguridad de canal utilizando TLS, consulte

- Protección de los canales con TLS

Para conocer detalles de los mandatos PCF asociados a TLS, consulte

- Cambiar, copiar y crear un objeto de información de autenticación
- Suprimir objeto de información de autenticación
- Consultar objeto de información de autenticación

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 194.

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 273.

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 198. 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 198.

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 199 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.

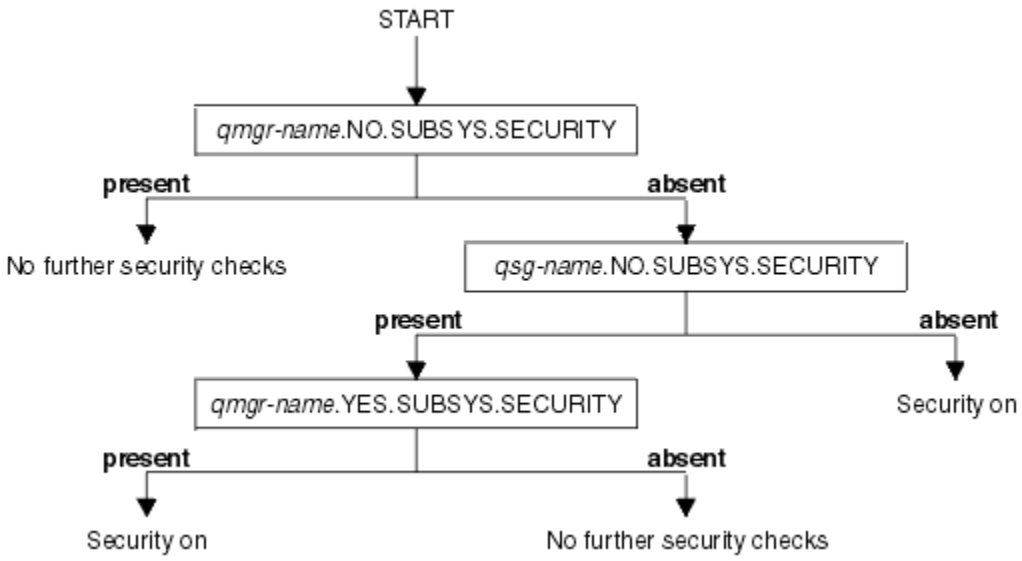


Figure 14. Checking for subsystem security

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 200 and Figure 16 on page 200 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.

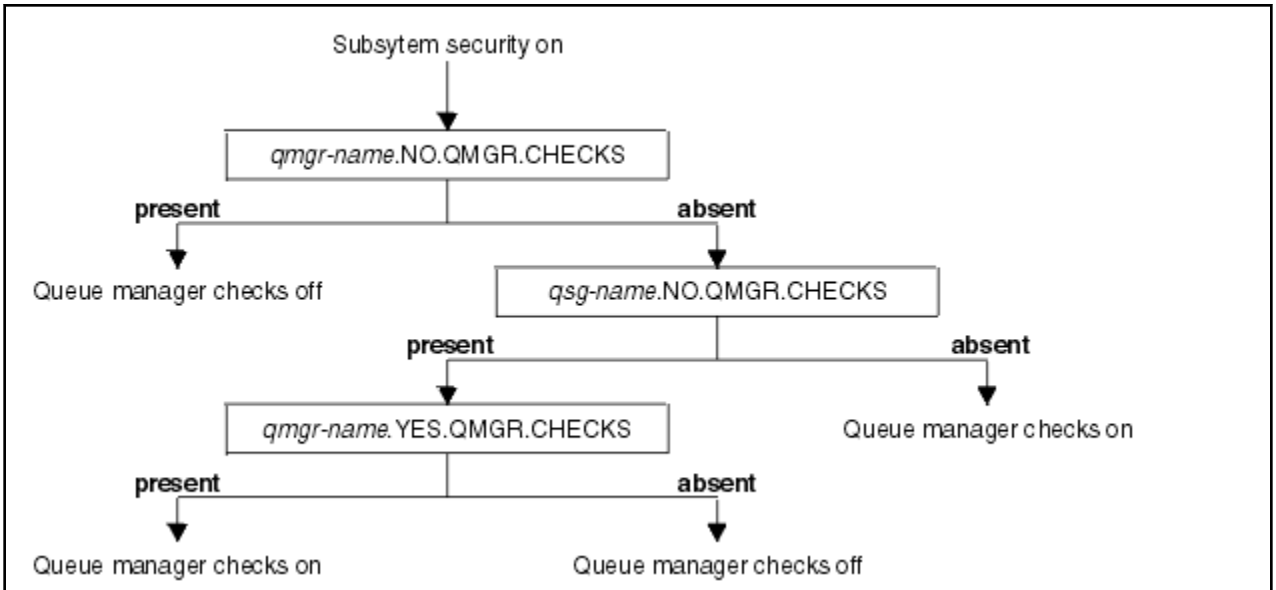


Figure 15. Checking for queue manager level security

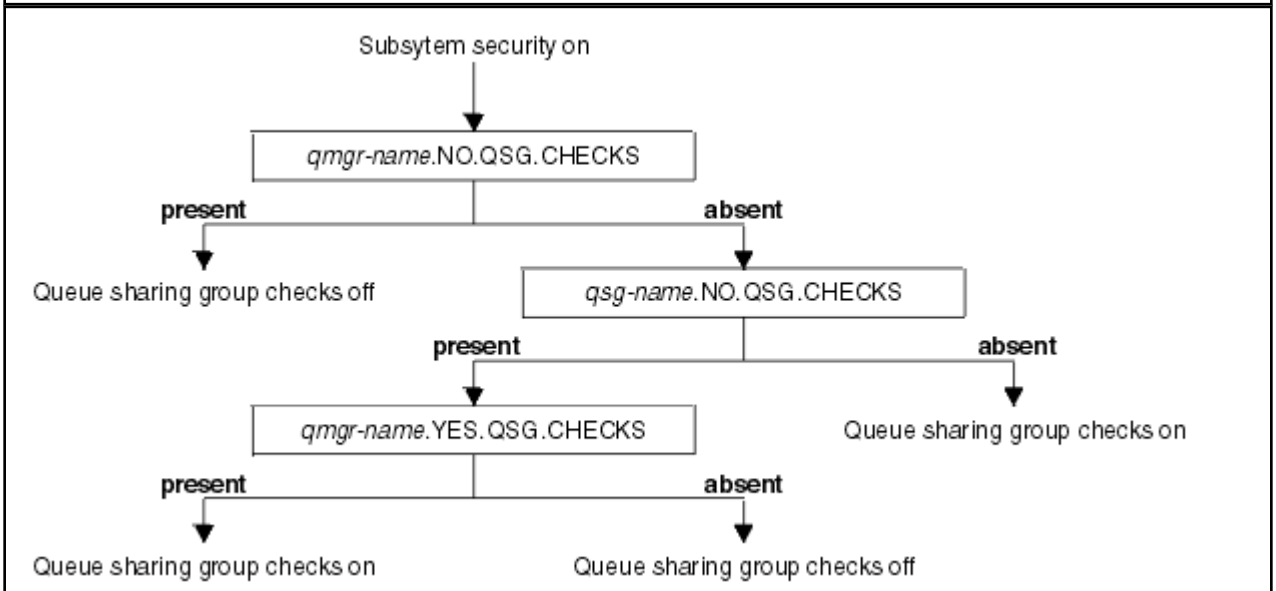


Figure 16. Checking for queue sharing group level security

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 200, Table 27 on page 201, Table 28 on page 201, and Table 29 on page 201 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 202 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 255 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 244.
2. Resource level security (RESLEVEL) checks are also made at connection time. For details, see “El perfil de seguridad RESLEVEL” on page 238.

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 210 and “Considerations for model queues” on page 211 .

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 224 and "Profiles for alternate user security" on page 222. 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 215.

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 244 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 “Seguridad de la mensajería remota” on page 106.

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 214 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 214.

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 215	-	-	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 220.

When you consider alias queues used for destination queues for PUT or GET restrictions, see “Considerations for alias queues” on page 210.

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 220.

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 222.

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 244. 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 215.

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 244 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 252.

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 247 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 208), and alternate user security (see "Profiles for alternate user security" on page 222). 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 215.

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 226.

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 227 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 232 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 232	hlq.ALTER.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
ALTER QMGR	hlq.ALTER.QMGR	ALTER	No check	-
ALTER QMODEL“5” on page 232	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 231	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 232	hlq.DEFINE.QLOCAL	ALTER	hlq.QUEUE.queue	ALTER
DEFINE QMODEL"5" on page 232	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 231	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 231	hlq.DISPLAY.CONN	READ	No check	-
DISPLAY GROUP	hlq.DISPLAY.GROUP	READ	No check	-
DISPLAY LOG “1” on page 231	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 231	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 231	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 231	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 231	-	-	-
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 "Seguridad de publicación/suscripción" on page 493
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 235	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 235	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 235	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 "Seguridad de publicación/suscripción" on page 493
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 235 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 236 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 El perfil de seguridad RESLEVEL

Puede definir un perfil especial en la clase MQADMIN o MXADMIN para controlar el número de identificadores (ID) de usuario que se comprueban para la seguridad de recursos de la API. Este perfil se denomina el perfil RESLEVEL. El modo en que este perfil afecta a la seguridad de recursos de la API depende de cómo se accede a IBM MQ.

Cuando una aplicación intenta conectarse a IBM MQ, IBM MQ comprueba el acceso que el ID de usuario asociado a la conexión tiene a un perfil en la clase MQADMIN o MXADMIN llamado:

```
hlq.RESLEVEL
```

Donde `hlq` puede ser `ssid` (ID del subsistema) o `qsg` (ID del grupo de compartición de colas).

Los ID de usuario asociados a cada tipo de conexión son:

- El ID de usuario de la tarea de conexión para las conexiones por lotes
- El ID de usuario del espacio de direcciones del CICS para conexiones CICS
- El ID de usuario del espacio de direcciones de la región IMSpara conexiones de IMS
- El ID de usuario del espacio de direcciones del iniciador de canal para las conexiones del iniciador de canal

Atención: RESLEVEL es una opción muy potente; puede hacer que se pasen por alto todas las comprobaciones de seguridad de recursos para una determinada conexión.

Si no tiene un perfil RESLEVEL definido, asegúrese de que ningún otro perfil de la clase MQADMIN coincida con hlq.RESLEVEL. Por ejemplo, si tiene un perfil en MQADMIN llamado hlq. * * y ningún perfil hlq.RESLEVEL , tenga cuidado con las consecuencias del hlq. * * porque se utiliza para la comprobación RESLEVEL.

Defina un perfil hlq.RESLEVEL y establezca el UACC en NONE, en vez de no tener perfil RESLEVEL alguno. Tenga el menor número de usuarios o grupos en la lista de acceso que sea posible. Para obtener información detallada sobre cómo auditar el acceso RESLEVEL, consulte “Auditing considerations on z/OS” en la página 263.

Si está utilizando solamente seguridad a nivel de gestor de colas, IBM MQ realiza comprobaciones RESLEVEL para el perfil `qmgr-name`.RESLEVEL. Si está utilizando solamente seguridad a nivel de grupo de compartición de colas, IBM MQ realiza comprobaciones RESLEVEL para el perfil `qsg-name`.RESLEVEL. Si está utilizando una combinación de seguridad a nivel de gestor de colas y de grupo de compartición de colas, IBM MQ comprueba primero la existencia de un perfil RESLEVEL a nivel de gestor de colas. Si no encuentra ninguno, busca un perfil RESLEVEL a nivel de grupo de compartición de colas.

Si no puede encontrar un perfil RESLEVEL, IBM MQ permite la comprobación del ID de trabajo y tarea (o de usuario alternativo) para una conexión CICS o IMS. Para una conexión por lotes, IBM MQ permite la comprobación del ID de usuario del trabajo (o alternativo). Para el iniciador de canal, IBM MQ permite la comprobación del ID de usuario de canal y el ID de usuario de MCA (o alternativo).

Si hay un perfil RESLEVEL, el nivel de comprobación depende del entorno y el nivel de acceso para el perfil.

Recuerde que si el gestor de colas es miembro de un grupo de compartición de colas y no define este perfil a nivel de gestor de colas, puede haber uno definido a nivel de grupo de compartición de colas que afectará al nivel de comprobación. Para activar la comprobación de dos ID de usuario, defina un perfil RESLEVEL (con el prefijo del nombre del gestor de colas del nombre del grupo de compartición de colas) con un UACC (NONE) y asegúrese de que los usuarios relevantes no tienen acceso otorgado a este perfil.

Cuando considere el acceso que el ID de usuario del iniciador del canal tiene a RESLEVEL, recuerde que la conexión establecida por el iniciador de canal es también la conexión utilizada por los canales. Un valor que hace que se pasen por alto todas las comprobaciones de seguridad de recursos para el ID de usuario del iniciador del canal omite de hecho las comprobaciones de seguridad para todos los canales. Si el ID de usuario del iniciador del canal es cualquiera que no sea NONE, entonces sólo se comprueba el acceso de un ID de usuario (para un nivel de acceso READ o UPDATE) o de ningún ID de usuario (para un nivel de acceso CONTROL o ALTER). si otorga al ID de usuario del iniciador de canal un nivel de acceso distinto de NONE a RESLEVEL, asegúrese de que comprende el efecto de este valor en las comprobaciones de seguridad realizadas para los canales.

La utilización del perfil RESLEVEL significa que no se toman registros de auditoría de seguridad normales. Por ejemplo, si pone UAUDIT en un usuario, no se realiza la auditoría del acceso al perfil hlq.RESLEVEL en MQADMIN.

Si utiliza la opción WARNING de RACF en el perfil hlq.RESLEVEL, no se generan mensajes de aviso RACF para los perfiles de la clase RESLEVEL.

Las comprobaciones de seguridad para los mensajes de informes como los COD los controla el perfil RESLEVEL asociado a la aplicación de origen. Por ejemplo, si un ID de usuario de trabajo por lotes tiene autorización de CONTROL o ALTER para un perfil RESLEVEL, entonces todas las comprobaciones

de recursos realizadas por el trabajo por lotes se omiten , incluida la comprobación de seguridad de los mensajes de informes.

Si cambia el perfil RESLEVEL, los usuarios deben desconectarse y conectarse de nuevo para que el cambio tenga efecto. (Esto incluye la detención y reinicio del iniciador de canal si se cambia el acceso que el ID de usuario del espacio de direcciones de gestión de colas distribuidas tiene al perfil RESLEVEL.)

Para desactivar la auditoría de RESLEVEL, utilice el parámetro del 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 240. 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 214, 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 241 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 247 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.

Table 55. Checks made at different RACF access levels for channel initiator connections (continued)	
RACF access level	**Level of checking**
UPDATE	Check one user ID.
CONTROL	No check.
ALTER	No check.
Note: See "User IDs used by the channel initiator" on page 247 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 251 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 251 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.

Issued from...	User ID contents
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.
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 241 in topic "RESLEVEL and CICS connections" on page 241, two user IDs are checked if the RESLEVEL profile is set to NONE. Then, from Table 58 on page 246 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.queuename profile**	**hlq.resourcename profile**
No, 1 check	-	REG	REG
No, 2 checks	-	REG+SEC	REG+SEC
Yes, 1 check	REG	REG	REG
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 247.

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
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

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**
DEF, 2 checks	-	CHL + MCA	CHL + MCA
CTX, 1 check	CHL	CHL	CHL
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)

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 "Control de accesos para clientes" on page 108 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 242 shows that two user IDs are checked because RESLEVEL is set to NONE.

Table 61 on page 248 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 256 summarizes the situations in which security information is cached and in which cached information is used.

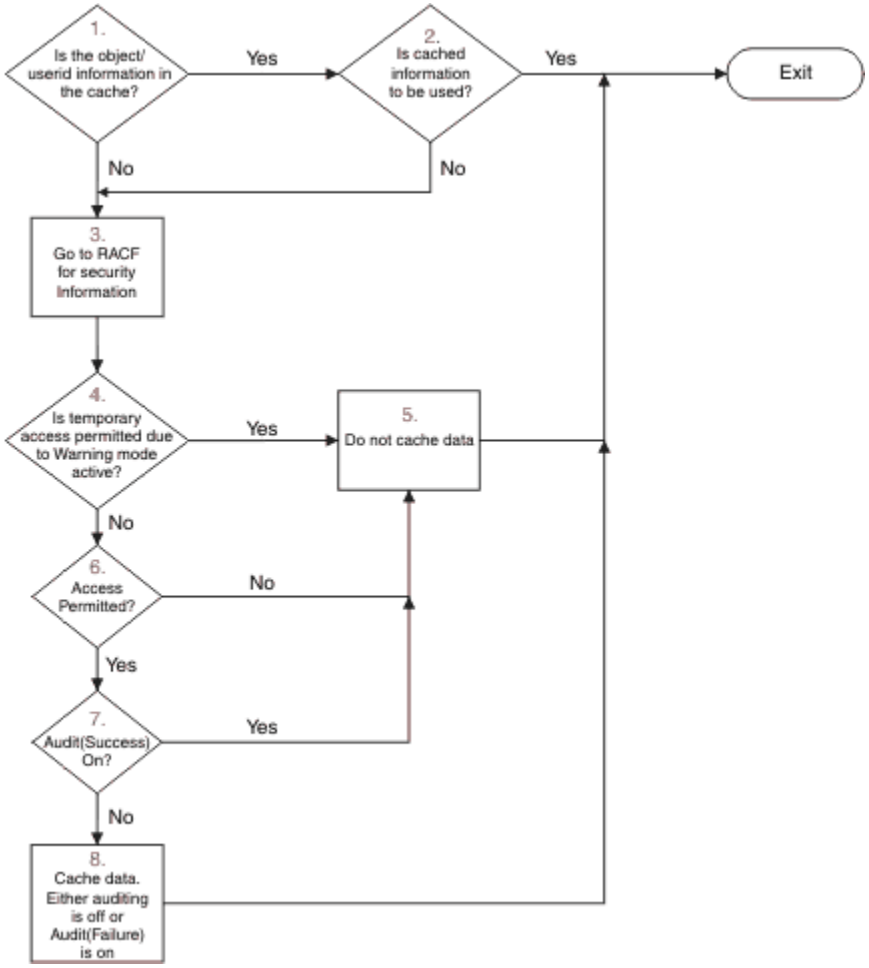


Figure 17. Logic flow for IBM MQ security caching

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 259 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 259 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 266, 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 ejercer un control más preciso sobre el acceso otorgado a la conexión de sistemas a nivel de canal, puede utilizar registros de autenticación 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 264.

```
      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 224.
- 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 214, and to all the user destination queues and the dead-letter queue (but see “Dead-letter queue security” on page 213).

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 224 "RESLEVEL and the channel initiator connection" on page 242 and "User IDs for security checking on z/OS" on page 244 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 207.

Data sets
The channel initiator address space needs appropriate access to queue manager data sets, see "Authorizing access to data sets" on page 259.

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 227.

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 244 for more information.

You can also use the Transport Layer Security (TLS) protocol to provide security on channels. See "Protocolos de seguridad TLS en IBM MQ" on page 25 for more information about using TLS with IBM MQ.

See also "Control de accesos para clientes" on page 108 for information about server-connection security.

User IDs
The user IDs described in "User IDs used by the channel initiator" on page 247 and "User IDs used by the intra-group queuing agent" on page 251 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 266:

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 227) 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 272.
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.

Configuración de la seguridad de IBM MQ MQI client

Debe tener en cuenta la seguridad del IBM MQ MQI client para que las aplicaciones cliente no tengan acceso sin restricciones a los recursos del servidor.

Al ejecutar una aplicación cliente, no ejecute la aplicación utilizando un ID de usuario que tenga más derechos de acceso que los necesarios; por ejemplo, un usuario en el grupo `mqm` o incluso el propio usuario `mqm`.

Al ejecutar una aplicación como un usuario con demasiados derechos de acceso, corre el riesgo de que la aplicación acceda y cambie partes del gestor de colas, ya sea de forma accidental o intencional.

Hay dos aspectos de seguridad entre una aplicación cliente y su servidor de gestor de colas: la autenticación y el control de accesos.

- La autenticación puede utilizarse para asegurarse de que la aplicación cliente, ejecutándose como un usuario específicos, sea quien dice ser. Utilizando la autenticación podrá impedir que un atacante acceda al gestor de colas suplantando una de sus aplicaciones:

  La autenticación se proporciona mediante una de las dos opciones siguientes:

  – La característica de autenticación de conexión.

    Para obtener más información sobre la autenticación de conexión, consulte “Autenticación de conexión” en la página 74.

  – Utilización de la autenticación mutua dentro de TLS.

    Para obtener más información sobre TLS, consulte “Trabajar con SSL/TLS” en la página 281.

- El control de acceso puede utilizarse para otorgar o eliminar derechos de acceso para un usuario específico o un grupo de usuarios. Si ejecuta una aplicación cliente con un usuario creado de forma específica (o usuario en un grupo específico) podrá utilizar los controles de acceso para asegurarse de que la aplicación no pueda acceder a partes del gestor de colas que la aplicación no debería.

  Al configurar el control de acceso deberá tener en cuenta las reglas de autenticación de canal y el campo MCAUSER en un canal. Ambas características tienen la capacidad de cambiar qué ID de usuario se está utilizando para verificar derechos de control de acceso.

  Para obtener más información sobre el control de acceso, consulte “Autorización del acceso a objetos” en la página 358.

Si ha configurado una aplicación cliente para conectarse a un canal específico con un ID restringido, pero el canal tiene un ID de administrador establecido en el campo MCAUSER, siempre y cuando la aplicación cliente se conecte satisfactoriamente, el ID de administrador se utilizará para las comprobaciones de control de acceso. Por lo tanto, la aplicación cliente tendrá derechos de acceso total al gestor de colas.

Para obtener más información sobre el atributo MCAUSER, consulte “Correlación de un ID de usuario cliente con un ID de usuario MCAUSER” en la página 394.

Las reglas de autenticación de canal también pueden utilizarse como método para controlar el acceso a un gestor de colas, estableciendo criterios y reglas específicas para que se acepte una conexión.

Para obtener más información sobre las reglas de autenticación de canal, consulte “Registros de autenticación de canal” en la página 53.

Especificación de que sólo se utilizan CipherSpecs certificadas por FIPS en el tiempo de ejecución del cliente MQI

Cree sus propios repositorios de claves utilizando software compatible con FIPS y, a continuación, especifique que el canal debe utilizar las CipherSpecs certificadas por FIPS.

Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC) . El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el certificado deIBM Crypto for C (ICC) y tener en cuenta cualquier consejo proporcionado por NIST. Un módulo FIPS 140-3 de sustitución está actualmente en curso y su estado se puede ver buscándolo en los módulos CMVP de NIST en la lista de procesos.

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar

buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los módulos CMVP de NIST en la lista de procesos.

Para que sean compatibles con FIPS en tiempo de ejecución, los repositorios de claves deben haberse creado y gestionado utilizando solo software compatible con FIPS como **runmqakm** con la opción `-fips` .

Puede especificar que un canal TLS debe utilizar sólo CipherSpecs certificadas por FIPS de tres maneras, listadas por orden de prioridad:

1. Establezca el campo FipsRequired de la estructura MQSCO en `MQSSL_FIPS_YES`.
2. Establezca la variable de entorno **MQSSLFIPS** en YES.
3. Establezca el atributo **SSLFipsRequired** en la stanza SSL del archivo de configuración de cliente en YES.

De forma predeterminada, las CipherSpecs certificadas por FIPS no son obligatorias.

Estos valores tienen el mismo significado que los valores de parámetro equivalentes en **ALTER QMGR SSLFIPS** (consulte **ALTER QMGR** (modificar valores del gestor de colas)). Si el proceso de cliente no tiene actualmente ninguna conexión TLS activa y se especifica un valor FipsRequired válido en una llamada MQCONNX de SSL, todas las conexiones TLS posteriores asociadas a este proceso deben utilizar únicamente las CipherSpecs asociadas a este valor. Esto se aplica hasta que ésta y el resto de conexiones TLS se hayan detenido, momento en el que una MQCONNX posterior puede proporcionar un nuevo valor para FipsRequired.

Si el hardware de cifrado está configurado, los módulos de cifrado que IBM MQ utiliza se pueden configurar con los módulos que proporciona el producto de hardware y estos pueden estar certificados por FIPS en un nivel determinado. Los módulos configurables y si tienen el certificado FIPS depende del producto de hardware que se utilice.

Siempre que sea posible, si se ha configurado CipherSpecs sólo para FIPS, el cliente MQI rechaza las conexiones que especifican una CipherSpec no FIPS con `MQRC_SSL_INITIALIZATION_ERROR`. IBM MQ no garantiza rechazar todas las conexiones de este tipo y es responsabilidad del usuario determinar si la configuración es compatible con IBM MQ.

Conceptos relacionados

“Federal Information Processing Standards (FIPS) para AIX, Linux, and Windows” en la página 36
Cuando la criptografía es necesaria en un canal SSL/TLS en sistemas AIX, Linux, and Windows , IBM MQ utiliza un paquete de criptografía denominado IBM Crypto for C (ICC). En las plataformas AIX, Linux, and Windows , el software ICC ha pasado el Programa de validación de criptomódulos FIPS (Federal Information Processing Standards) del Instituto Nacional de Estándares y Tecnología de Estados Unidos, en el nivel 140-2.

AIX Ejecución de aplicaciones cliente TLS con varias instalaciones de GSKit 8.0 en AIX

Las aplicaciones cliente TLS en AIX podrían experimentar `MQRC_CHANNEL_CONFIG_ERROR` y error `AMQ6175` al ejecutarse en sistemas AIX con varias instalaciones de IBM Global Security Kit (GSKit) 8.0 .

Al ejecutar aplicaciones cliente en un sistema AIX con varias instalaciones de GSKit 8.0 , las llamadas de conexión de cliente pueden devolver `MQRC_CHANNEL_CONFIG_ERROR` cuando se utiliza TLS. Los registros de `/var/mqm/errors` registran el error `AMQ6175` y `AMQ9220` para la aplicación cliente anómala, por ejemplo:

```
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 ----------------------------------------------------------
```

Una causa común de este error es que el valor de la variable de entorno LIBPATH o LD_LIBRARY_PATH ha hecho que el cliente IBM MQ cargue un conjunto mixto de bibliotecas de dos instalaciones de GSKit 8.0 diferentes. La ejecución de una aplicación cliente IBM MQ en un entorno de Db2 puede producir este error.

Para evitarlo, incluya los directorios de la biblioteca de IBM MQ en la parte frontal de la vía de acceso de bibliotecas para que las bibliotecas de IBM MQ tengan prioridad. Esto se puede lograr utilizando el mandato **setmqenv** con el parámetro **-k** , por ejemplo:

```
. /usr/mqm/bin/setmqenv -s -k
```

Para obtener más información sobre el uso del mandato **setmqenv**, consulte setmqenv (establezca el entorno de IBM MQ)

Configuración de canales TLS con MQSC

Para configurar canales TLS, utilice los mandatos **runmqsc** y ALTER CHANNEL. Existe la opción de configurar un canal para que sólo acepte certificados que tengan atributos en el nombre distinguido del propietario que coincidan con los valores dados. También puede configurar opcionalmente un canal del gestor de colas para que el gestor rehúse la conexión si la parte iniciadora no envía su propio certificado personal.

Acerca de esta tarea

Para configurar canales en IBM MQ Explorer, consulte Configuración de canales TLS con IBM MQ Explorer.

Para configurar canales utilizando **runmqsc**, realice los pasos siguientes.

Procedimiento

1. Invoque el mandato **runmqsc** conectándose al gestor de colas de destino.
2. Identifique el canal que desea habilitar para TLS.
   Anote el nombre de canal y el tipo de canal.
3. Utilice el mandato ALTER CHANNEL para modificar varias propiedades de un canal IBM MQ .

Proporcione el nombre de canal y el tipo de canal además del mandato. Por ejemplo, para modificar un canal emisor denominado MQ.TEST ejecute el mandato siguiente:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR)
```

Existen diversos atributos de canal relacionados con TLS que puede ajustar en las definiciones de canal de IBM MQ .

Qué hacer a continuación

Definición de la Seguridad de mensajes

La mensajería habilitada para TLS ofrece dos métodos para garantizar la seguridad de los mensajes:

- El cifrado asegura que si el mensaje es interceptado, no podrá leerse.
- Las funciones hash aseguran que si el mensaje se modifica, esta acción se detecta.

La combinación de estos métodos se denomina especificación de cifrado o CipherSpec. Se debe definir la misma CipherSpec para ambos extremos de un canal, de lo contrario la mensajería habilitada para TLS falla. Para obtener más información, consulte "Protección de IBM MQ" en la página 7.

Para modificar un canal IBM MQ habilite TLS, especifique un valor en el atributo SSLCIPH. Este atributo debe establecerse en una CipherSpec válida para la plataforma de colas del gestor de colas de la lista "Habilitación de CipherSpecs" en la página 428.

Para modificar un canal IBM MQ para inhabilitar TLS, establezca SSLCIPH en un valor en blanco. Por ejemplo:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCIPH(ANY_TLS12_OR_HIGHER)
```

Nota: Debe incluir el nombre de canal entre comillas simples para asegurarse de que se mantienen las mayúsculas y minúsculas del carácter. Sin comillas simples, IBM MQ transforma la serie para que esté en mayúsculas.

Filtrado de certificados en nombre de su propietario

Los certificados contienen el nombre distinguido del propietario del certificado. Existe la opción de configurar el canal para que sólo acepte certificados que tengan atributos en el nombre distinguido del propietario que coincidan con los valores dados.

Los nombres de atributo que puede filtrar IBM MQ aparecen en la tabla siguiente:

Nombres de atributo	**Significado**
SERIALNUMBER	Número de serie de certificado
MAIL	Dirección de correo electrónico
Deprecated E	Dirección de correo electrónico (En desuso por ser preferible MAIL)
UID o USERID	Identificador de usuario
CN	Nombre común
T	Título
OU	Nombre de la unidad organizativa
DC	Componente de dominio
O	Nombre de la organización
CALLE	Calle / Primera línea de dirección
L	Nombre de la localidad

Nombres de atributo	Significado
ST (o SP o S)	Nombre del estado o provincia
PC	Código postal
C	País
UNSTRUCTUREDNAME	Nombre de host
UNSTRUCTUREDADDRESS	Dirección IP
DNQ	Calificador de nombre distinguido

Puede utilizar el carácter comodín (*) al principio o al final del valor de atributo en lugar de cualquier número de caracteres. Por ejemplo, para aceptar sólo certificados de personas que se apelliden `Smith` y que trabajen para `IBM` en `GB`, escriba:

```
CN=*Smith, O=IBM, C=GB
```

Por ejemplo:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLPEER('CN=*Smith, O=IBM, C=GB')
```

Nota: Debe encerrar la serie SSLPEER entre comillas simples para asegurarse de que se mantienen las mayúsculas y minúsculas del carácter. Sin comillas simples, IBM MQ transforma la serie para que esté en mayúsculas.

Autenticación de entidades que inician conexiones con un gestor de colas

Cuando otra parte inicie una conexión habilitada para TLS con un gestor de colas, el gestor de colas debe enviar su certificado personal a la parte iniciadora como prueba de la identidad. También puede configurar opcionalmente el canal del gestor de colas para que el gestor rehúse la conexión si la parte iniciadora no envía su propio certificado personal.

Para ello, establezca el atributo SSLCAUTH. Este atributo es un atributo booleano y puede tener los valores OPTIONAL o REQUIRED:

- OPTIONAL autentica el certificado de un cliente que se conecta si se proporciona uno, pero no requiere que un cliente envíe uno. Un cliente se rechaza si envía un certificado que no es válido.
- REQUIRED rechaza cualquier cliente de conexión que no proporcione un certificado TLS válido

Por ejemplo:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCAUTH(REQUIRED)
```

IBM i Configuración de las comunicaciones para SSL o TLS en IBM i

Las comunicaciones seguras que utilizan los protocolos de seguridad de cifrado SSL o TLS comportan la configuración de canales de comunicación y la gestión de los certificados digitales que utilizará para la autenticación.

Para configurar la instalación de SSL o TLS, debe definir los canales para que utilicen SSL o TLS. También debe crear y gestionar los certificados digitales. En algunos sistemas operativos, puede realizar las pruebas con certificados autofirmados. Sin embargo, en IBM i, debe utilizar certificados personales firmados por una CA local.

Para obtener información completa sobre la creación y gestión de certificados, consulte "Trabajar con SSL/TLS en IBM i" en la página 281.

En esta colección de temas se presentan algunas de las tareas que forman parte de la configuración de las comunicaciones SSL o TLS, y se proporciona una guía paso a paso para completar estas tareas.

Es posible que también desee probar la autenticación de cliente SSL o TLS, que son partes opcionales de los protocolos SSL y TLS. Durante el reconocimiento SSL o TLS, el cliente TLS o SSL siempre obtiene y valida un certificado digital del servidor. Con la implementación de IBM MQ, el servidor SSL o TLS siempre solicita un certificado del cliente.

En IBM i, el cliente SSL o TLS solo envía un certificado si tiene uno etiquetado con el formato de IBM MQ correcto:

- Para un gestor de colas, `ibmwebspheremq` seguido del nombre del gestor de colas en minúsculas. Por ejemplo, para `QM1`, `ibmwebspheremqqm1`.
- Para un cliente C de IBM MQ para IBM i, `ibmwebspheremq` seguido del ID de usuario de inicio de sesión cambiado a minúsculas, por ejemplo, `ibmwebspheremqmyuserid`.

IBM MQ utiliza el prefijo `ibmwebspheremq` en una etiqueta para evitar confusiones con certificados de otros productos. Asegúrese de especificar la etiqueta completa del certificado en minúsculas.

El servidor SSL o TLS siempre valida el certificado de cliente si se envía uno. Si el cliente SSL o TLS no envía un certificado, la autenticación falla solo si el extremo del canal que actúa como servidor SSL o TLS se define con el parámetro SSLCAUTH establecido en REQUIRED o un valor de parámetro SSLPEER establecido. Para obtener más información, consulte Conexión de dos gestores de colas utilizando SSL o TLS.

ALW Configuración de las comunicaciones para SSL o TLS en AIX, Linux, and Windows

Las comunicaciones seguras que utilizan los protocolos de seguridad de cifrado SSL o TLS comportan la configuración de canales de comunicación y la gestión de los certificados digitales que utilizará para la autenticación.

Para configurar la instalación de SSL o TLS, debe definir los canales para que utilicen SSL o TLS. También debe crear y gestionar los certificados digitales. En sistemas AIX, Linux, and Windows , puede realizar las pruebas con certificados autofirmados.

Atención: No es posible utilizar una combinación de certificados firmados por Elliptic Curve y los certificados firmados por RSA en los gestores de colas que desea unir utilizando los canales habilitados para TLS.

Los gestores de colas que utilizan los canales habilitados para TLS deben utilizar todos los certificados firmados por RSA, o bien todos los certificados firmados por EC, no una combinación de ambos.

Consulte “Certificados digitales y compatibilidad de CipherSpec en IBM MQ” en la página 49 para obtener más información.

Los certificados autofirmados no se pueden revocar, lo que podría permitir a un atacante suplantar una identidad después de que se haya comprometido una clave privada. Las CA pueden revocar un certificado comprometido, lo que impide su posterior uso. Los certificados firmados por una CA son, por lo tanto, más seguros para su uso en un entorno de producción, aunque los certificados autofirmados son más convenientes para un sistema de prueba.

Para obtener información completa sobre la creación y gestión de certificados, consulte “Trabajar con SSL/TLS en AIX, Linux, and Windows” en la página 299.

En esta colección de temas se presentan algunas de las tareas que forman parte de la configuración de las comunicaciones SSL y se proporciona una guía paso a paso para completar estas tareas.

Es posible que también desee probar la autenticación de cliente SSL o TLS, que es una parte opcional de los protocolos. Durante el reconocimiento SSL o TLS, el cliente TLS o SSL siempre obtiene y valida un certificado digital del servidor. Con la implementación de IBM MQ, el servidor SSL o TLS siempre solicita un certificado del cliente.

En AIX, Linux, and Windows, el cliente SSL o TLS solo envía un certificado si tiene uno etiquetado con el formato de IBM MQ correcto:

- Para un gestor de colas, el formato es `ibmwebspheremq` seguido del nombre del gestor de colas que ha cambiado a minúsculas. Por ejemplo, para `QM1`, `ibmwebspheremqqm1`
- Para un cliente de IBM MQ , `ibmwebspheremq` seguido por el ID de usuario de inicio de sesión cambiado a minúsculas, por ejemplo, `ibmwebspheremqmyuserid`.

IBM MQ utiliza el prefijo `ibmwebspheremq` en una etiqueta para evitar confusiones con certificados de otros productos. Asegúrese de especificar la etiqueta completa del certificado en minúsculas.

El servidor SSL o TLS siempre valida el certificado de cliente si se envía uno. Si el cliente no envía un certificado, la autenticación falla sólo si el extremo del canal que actúa como servidor SSL o TLS se define con el parámetro SSLCAUTH establecido en REQUIRED o un valor de parámetro SSLPEER establecido. Para obtener más información, consulte Conexión de dos gestores de colas utilizando SSL o 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 314.

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.

Trabajar con SSL/TLS

Estos temas proporcionan instrucciones para realizar tareas individuales relacionados con la utilización de TLS con IBM MQ.

Muchos de ellos se utilizan como pasos de las tareas de nivel superior que se describen en los apartados siguientes:

- "Identificación y autenticación de usuarios" en la página 325
- "Autorización del acceso a objetos" en la página 358
- "Confidencialidad de mensajes" en la página 428
- "Integridad de datos de mensajes" en la página 485
- "Mantenimiento de la seguridad de los clústeres" en la página 486

IBM i Trabajar con SSL/TLS en IBM i

Esta colección de temas ofrece instrucciones de tareas individuales relacionadas con TLS (Seguridad de la capa de transporte) en IBM MQ for IBM i.

Para IBM i, el soporte para TLS forma parte integral del sistema operativo. Asegúrese de que ha instalado los requisitos previos que se listan en Requisitos de hardware y software en IBM i.

En IBM i, las claves y los certificados digitales se gestionan con la herramienta Gestor de certificados digitales (DCM).

Acceso a DCM

Siga estas instrucciones para acceder a la interfaz del DCM.

Acerca de esta tarea

Realice los pasos siguientes en un navegador web que admita marcos.

Procedimiento

1. Vaya a `http://`*`machine`*`.domain:2001` o `https://`*`machine`*`.domain:2010`, donde *máquina* es el nombre del sistema.
2. Escriba un perfil de usuario y una contraseña válidos cuando se le solicite.

   Asegúrese de que el perfil de usuario tenga las autorizaciones especiales *ALLOBJ y *SECADM que le permitan crear almacenes de certificados nuevos. Si no tiene las autorizaciones especiales, sólo puede gestionar los certificados personales o ver las firmas de objetos correspondientes a los objetos para los que tiene autorización. Si tiene autorización para utilizar una aplicación de firma de objetos, también puede firmar objetos desde DCM.
3. En la página Configuraciones de Internet, pulse **Gestor de certificados digitales**.

   Se visualiza la página Gestor de certificados digitales.

Asignación de un certificado a un gestor de colas en IBM i

Utilice DCM para asignar un certificado a un gestor de consultas.

Utilice una gestión de certificados digitales de IBM i tradicional para asignar un certificado a un gestor de colas. Esto significa que puede especificar que un gestor de colas utilice el almacén de certificados del sistema, y que el gestor de colas se registre para su uso como aplicación con el Gestor de certificados digitales (DCM). Para ello, cambie el valor del atributo **SSLKEYR** del gestor de colas a *SYSTEM.

Cuando el parámetro **SSLKEYR** se cambia a *SYSTEM, IBM MQ registra el gestor de colas como una aplicación de servidor con una etiqueta de aplicación exclusiva de QIBM_WEBSPHERE_MQ_QMGRNAME y una etiqueta con una descripción de Qmgrname (WMQ). Tenga en cuenta que los atributos **CERTLABL** del canal no se utilizan si utiliza el almacén de certificados *SYSTEM. El gestor de colas aparece entonces como una aplicación de servidor en el Gestor de certificados digitales, y puede asignar a esta aplicación cualquier certificado de servidor o de cliente en el almacén de certificados.

Como el gestor de colas está registrado como una aplicación, se pueden efectuar las funciones avanzadas de DCM, como la definición de listas de CA fiables.

Si el parámetro **SSLKEYR** se cambia a un valor distinto de *SYSTEM, IBM MQ anula el registro del gestor de colas como una aplicación con el Certificate Managerdigital. Si se suprime un gestor de colas, también se anula el registro del DCM. Un usuario con autorización *SECADM suficiente también puede añadir o eliminar manualmente aplicaciones de DCM.

Configuración de un repositorio de claves en IBM i

Se debe configurar un depósito de claves en ambos extremos de la conexión. Se pueden utilizar los almacenes de certificados predeterminados o puede crear el suyo propio.

Una conexión TLS requiere un *depósito de claves* en cada extremo de la conexión. Cada gestor de colas y IBM MQ MQI client debe tener acceso a un repositorio de claves. Si desea acceder al repositorio de claves utilizando un nombre de archivo y contraseña (es decir, sin utilizar la opción *SYSTEM), asegúrese de que el perfil de usuario QMQM tiene las autorizaciones siguientes:

- Autorización de ejecución para el directorio que contiene el depósito de claves
- Autorización de lectura para el archivo que contiene el depósito de claves

Consulte “Repositorio de claves SSL/TLS” en la página 26 para obtener más información. Tenga en cuenta que los atributos de canal **CERTLABL** no se utilizan si utiliza el almacén de certificados *SYSTEM.

En IBM i, los certificados digitales se almacenan en un almacén de certificados que se gestiona con DCM. Estos certificados digitales tienen etiquetas, que asocian el certificado con un gestor de colas o un IBM MQ MQI client. TLS utiliza los certificados con fines de autenticación.

La etiqueta es el valor del atributo **CERTLABL**, si éste está establecido o el valor predeterminado `ibmwebspheremq` con el nombre del gestor de colas o el ID de usuario de inicio de sesión de IBM MQ MQI client añadido, todo en minúsculas. Para obtener información detallada, consulte Etiquetas de certificados digitales.

El nombre del almacén de certificados del IBM MQ MQI client o gestor de colas consta de una vía de acceso y un nombre de raíz. La vía de acceso predeterminada es `/QIBM/UserData/ICSS/Cert/Server/` y el nombre de raíz predeterminado es `Default`. En IBM i, el almacén de certificados predeterminado, `/QIBM/UserData/ICSS/Cert/Server/Default.kdb`, también se conoce como `*SYSTEM`. Opcionalmente, puede definir su propia vía de acceso y nombre de raíz.

Si define su propia vía de acceso o nombre de archivo, establezca los permisos del archivo para controlar estrechamente el acceso al mismo.

En el apartado “Cambiar la ubicación del repositorio de claves para un gestor de colas en IBM i” en la página 285 se indica cómo especificar el nombre de almacén de certificados. Puede especificar el nombre del almacén de certificados antes o después de crear el almacén de certificados.

Nota: Las operaciones que puede realizar con DCM pueden estar limitadas por la autorización de su perfil de usuario. Por ejemplo, necesita las autorizaciones *ALLOBJ y *SECADM para crear un certificado de CA.

IBM i *Cifrado de contraseñas de repositorio de claves en IBM i*

Varios componentes de IBM MQ necesitan acceso a un repositorio de claves que contenga certificados digitales o claves simétricas. Un repositorio de claves está protegido con una contraseña, ya que contiene información confidencial. La contraseña del repositorio de claves debe almacenarse en una ubicación en la que IBM MQ pueda leerla cuando se acceda al repositorio de claves. La contraseña también debe estar cifrada para reducir la probabilidad de acceso no autorizado al repositorio de claves.

Los siguientes componentes y características de IBM MQ dan soporte a dos métodos diferentes para almacenar contraseñas de repositorio de claves:

- El repositorio de claves TLS del gestor de colas.
- IBM MQ MQI clients que utilizan TLS.

Las contraseñas de repositorio de claves para que las utilicen estos componentes se protegen utilizando el sistema de protección de contraseñas de IBM MQ . El mecanismo para proporcionar una contraseña y cifrarla varía ligeramente en función del componente:

El repositorio de claves TLS del gestor de colas

La contraseña se cifra cuando el atributo del gestor de colas **SSLKEYRPWD** se establece utilizando el mandato CHGMQM (Cambiar gestor de colas de mensajes) .

La contraseña se cifra con el algoritmo AES-128 . Los detalles de este algoritmo son de conocimiento público y se considera seguro.

La contraseña se almacena en un archivo de ocultación en un formato propietario que no entiende otro software que pueda acceder al repositorio de claves.

Una contraseña cifrada por un componente IBM MQ no puede ser utilizada por un componente IBM MQ diferente.

Se puede proporcionar una clave de cifrado exclusiva cuando la contraseña del repositorio de claves está cifrada. Una clave de cifrado exclusiva impide que cualquier persona que no tenga acceso a la clave de cifrado pueda descifrar la contraseña. Proporcione esta clave a través del atributo de gestor de colas **INITKEY** , que se debe establecer antes de proporcionar una contraseña que se va a cifrar.

Para obtener más información sobre el sistema de protección por contraseña de IBM MQ , consulte “Protección de contraseñas en archivos de configuración de componentes de IBM MQ” en la página 575.

IBM MQ MQI clients que utilizan TLS

“Programa de utilidad de Cliente SSL de IBM MQ (amqrsslc) para IBM i” en la página 297 puede almacenar la contraseña del repositorio de claves en un archivo de ocultación. Consulte también Administración utilizando mandatos MQSC en IBM i.

La contraseña se cifra con el algoritmo AES-128 . Los detalles de este algoritmo son de conocimiento público y se considera seguro.

La contraseña se almacena en un archivo de ocultación en un formato propietario que no entiende otro software que pueda acceder al repositorio de claves.

Se puede proporcionar una clave de cifrado exclusiva cuando la contraseña del repositorio de claves está cifrada. Una clave de cifrado exclusiva impide que cualquier persona que no tenga acceso a la clave de cifrado pueda descifrar la contraseña. Proporcione esta clave a través del parámetro **-sf** .

La contraseña cifrada se almacena en un archivo de ocultación en el mismo directorio que el archivo de repositorio de claves.

IBM MQ MQI clients también da soporte a las contraseñas proporcionadas a través de otros mecanismos. Consulte “Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en IBM i” en la página 287.

Independientemente del método que elija para cifrar la contraseña del repositorio de claves, asegúrese de que conoce las limitaciones del cifrado de contraseñas almacenadas. Consulte “Los límites de la protección a través del cifrado de contraseña” en la página 583.

Conceptos relacionados

“Suministro de la contraseña del repositorio de claves para un gestor de colas en IBM i” en la página 286
Puesto que el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

“Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en IBM i” en la página 287
Puesto que el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

“Trabajar con SSL/TLS en IBM i” en la página 281

Esta colección de temas ofrece instrucciones de tareas individuales relacionadas con TLS (Seguridad de la capa de transporte) en IBM MQ for IBM i.

Crear un almacén de certificados en IBM i
Si no desea utilizar el almacén de certificados predeterminado, siga este procedimiento para crear el suyo propio.

Acerca de esta tarea

Cree un almacén de certificados nuevo sólo si no desea utilizar el almacén de certificados predeterminado de IBM i.

Para especificar que se va a utilizar el almacén de certificados del sistema IBM i , cambie el valor del atributo SSLKEYR del gestor de colas a *SYSTEM. Este valor indica que el gestor de colas utiliza el almacén de certificados del sistema, y el gestor de colas se registra para su uso como aplicación con el Gestor de certificados digitales (DCM).

Procedimiento

1. Acceda a la interfaz DCM, tal y como se describe en "Acceso a DCM" en la página 281
2. En el panel de navegación, pulse **Crear almacén de certificados nuevo**.
   La página Crear almacén de certificados nuevo se visualiza en la sección de tareas.
3. En la sección de tareas, seleccione **Otro almacén de certificados del sistema** y pulse **Continuar**.
   Se visualiza la página Crear un certificado en almacén de certificados nuevo en la sección de tareas.
4. Seleccione **No - No crear un certificado en el almacén de certificados** y pulse **Continuar**.
   La página Nombre y contraseña de almacén de certificados se visualiza en la sección de tareas.
5. En el campo **Vía de acceso de almacén de certificados y nombre de archivo** , escriba una vía de acceso y un nombre de archivo de IFS, por ejemplo `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`
6. Escriba una contraseña en el campo **Contraseña** y vuélvala a escribir en el campo **Confirmar contraseña**. Pulse **Continuar**.
   Anote la contraseña (que es sensible a mayúsculas y minúsculas) porque la necesitará cuando oculte la clave del repositorio.
7. Para salir del DCM, cierre la ventana del navegador.

Qué hacer a continuación

Cuando haya creado el almacén de certificados mediante DCM, asegúrese de ocultar la contraseña, tal como se describe en "Ocultación de la contraseña del almacén de certificados en sistemas IBM i" en la página 284

Tareas relacionadas

"Importar un certificado a un repositorio de claves en IBM i" en la página 295
Siga este procedimiento para importar un certificado.

Ocultación de la contraseña del almacén de certificados en sistemas IBM i
Oculte la contraseña del almacén de certificados utilizando mandatos CL.

Las siguientes instrucciones se aplican a la ocultación de la contraseña del almacén de certificados en IBM i para un gestor de colas. De forma alternativa, para un IBM MQ MQI client, si no está utilizando el almacén de certificados *SYSTEM (es decir, el entorno MQSSLKEYR está establecido en un valor distinto de *SYSTEM), siga el procedimiento descrito en la sección "Ocultar la contraseña del almacén de certificados" en la página 298 de "Programa de utilidad de Cliente SSL de IBM MQ (amqrsslc) para IBM i" en la página 297.

Si ha especificado que debe utilizarse el almacén de certificados *SYSTEM (cambiando el valor del atributo SSLKEYR del gestor de colas a *SYSTEM), no debe seguir estos pasos.

Cuando haya creado el almacén de certificados mediante DCM, utilice los mandatos siguientes para ocultar la contraseña:

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

La contraseña distingue entre mayúsculas y minúsculas. Debe especificarse entre comillas simples exactamente como lo ha hecho en el paso 6 de “Crear un almacén de certificados en IBM i” en la página 284.

Nota: Si no utiliza el almacén de certificados del sistema predeterminado, y no oculta la contraseña, los intentos de iniciar los canales TLS no se ejecutarán correctamente porque éstos no podrán obtener la contraseña necesaria para acceder al almacén de certificados.

Protección por contraseña

Cuando se especifica una contraseña de repositorio de claves, IBM MQ cifra la contraseña utilizando el sistema IBM MQ Password Protection. Para cifrar la contraseña se utiliza una clave inicial; si no se proporciona al gestor de colas, se utiliza en su lugar una clave predeterminada.

Antes de proporcionar la contraseña del repositorio de claves, debe establecer una clave inicial exclusiva para el gestor de colas. Puede hacerlo utilizando el atributo **INITKEY** del mandato MQSC **ALTER QMGR** :

```
ALTER QMGR INITKEY('value')
```

Localizar el repositorio de claves para un gestor de colas en IBM i

Utilice este procedimiento para obtener la ubicación del almacén de certificados del gestor de colas.

Procedimiento

1. Visualice los atributos del gestor de colas, utilizando el siguiente mandato:

   ```
   DSPMQM MQMNAME('queue manager name')
   ```

2. Examine la salida del mandato para localizar la vía de acceso y nombre de raíz del almacén de certificados.

   Por ejemplo: `/QIBM/UserData/ICSS/Cert/Server/Default`, donde `/QIBM/UserData/ICSS/Cert/Server` es la vía de acceso y `Default` es el nombre de raíz.

Cambiar la ubicación del repositorio de claves para un gestor de colas en IBM i

Cambie la ubicación del almacén de certificados del gestor de colas utilizando CHGMQM o ALTER QMGR.

Procedimiento

Utilice el mandato CHGMQM o el mandato MQSC ALTER QMGR para establecer el atributo de repositorio de claves del gestor de colas.

a) Utilización de CHGMQM: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

b) Utilización de ALTER QMGR: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

En ambos casos, el almacén de certificados tiene el nombre de archivo completo: `/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb`

Qué hacer a continuación

Cuando cambia la ubicación del almacén de certificados de un gestor de colas, los certificados no se transfieren desde la ubicación antigua. Si los certificados de CA que se instalaron de antemano al crear el almacén de certificados son insuficientes, debe llenar el nuevo almacén de certificados con certificados, tal como se describe en “Importar un certificado a un repositorio de claves en IBM i” en la página 295.

También debe ocultar la contraseña para la nueva ubicación, según se describe en "Ocultación de la contraseña del almacén de certificados en sistemas IBM i" en la página 284.

IBM i *Suministro de la contraseña del repositorio de claves para un gestor de colas en IBM i*

Puesto que el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

IBM MQ proporciona un mecanismo para proporcionar la contraseña del repositorio de claves a un gestor de colas:

- El parámetro **SSLKEYRPWD** en el mandato **CHGMQM**

La contraseña del repositorio de claves se cifra utilizando el sistema de protección de contraseñas de IBM MQ . Para obtener más información sobre los métodos de protección de la contraseña del repositorio de claves, consulte "Cifrado de contraseñas de repositorio de claves en IBM i" en la página 282.

Consulte también Administración utilizando mandatos MQSC en IBM i.

El atributo SSLKEYRPWD

Para proporcionar una contraseña de repositorio de claves directamente al gestor de colas, ejecute el siguiente mandato **CHGMQM** , sustituyendo *queue_manager* por el nombre del gestor de colas y *password* por la contraseña del repositorio de claves.

```
CHGMQM MQMNAME('queue_manager') SSLKEYRPWD('password')
```

Atención: Asegúrese de rodear el nombre y la contraseña del gestor de colas con comillas simples; de lo contrario, IBM MQ convertirá los caracteres a mayúsculas.

Cuando se especifica una contraseña de repositorio de claves utilizando este método, la contraseña se cifra utilizando el sistema de protección de contraseñas de IBM MQ antes de que se almacene.

Una clave de cifrado, que se conoce como la clave inicial, se utiliza para cifrar la contraseña. Establezca el gestor de colas para que utilice una clave inicial exclusiva para proteger de forma segura la contraseña. Si no proporciona una clave inicial, se utiliza la clave predeterminada.

Asegúrese de que el gestor de colas esté configurado con una clave inicial exclusiva antes de establecer la contraseña del repositorio de claves. Puede modificar la clave inicial utilizando el atributo **INITKEY** en el mandato **ALTER QMGR** . Por ejemplo:

```
ALTER QMGR INITKEY('mykey')
```

Aviso: Si modifica la clave inicial después de establecer la contraseña del repositorio de claves, la contraseña del repositorio de claves no se cifra con la nueva clave inicial. Si cambia la clave inicial, también debe restablecer la contraseña del repositorio de claves. De lo contrario, IBM MQ no puede descifrar la contraseña del repositorio de claves y, por lo tanto, no puede acceder al repositorio de claves.

Para obtener más información sobre el atributo **SSLKEYRPWD** , consulte El parámetro **SSLKEYRPWD** en el mandato **CHGMQM**.

Conceptos relacionados

"Cifrado de contraseñas de repositorio de claves en IBM i" en la página 282
Varios componentes de IBM MQ necesitan acceso a un repositorio de claves que contenga certificados digitales o claves simétricas. Un repositorio de claves está protegido con una contraseña, ya que contiene información confidencial. La contraseña del repositorio de claves debe almacenarse en una ubicación en la que IBM MQ pueda leerla cuando se acceda al repositorio de claves. La contraseña también debe estar cifrada para reducir la probabilidad de acceso no autorizado al repositorio de claves.

"Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en IBM i" en la página 287

Puesto que el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

IBM i *Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en IBM i*

Puesto que el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

IBM MQ proporciona cuatro mecanismos para proporcionar la contraseña del repositorio de claves a un IBM MQ MQI client:

- "Los campos KeyRepoPassword de MQSCO " en la página 287
- "La variable de entorno MQKEYRPWD" en la página 288
- "El atributo SSLKeyRepositoryPassword del archivo de configuración del cliente" en la página 288
- "El archivo de ocultación del repositorio de claves" en la página 288

Si no utiliza un archivo de ocultación de repositorio de claves, puede proporcionar la contraseña del repositorio de claves como una serie de texto sin formato, o una serie que se cifre utilizando el sistema de protección de contraseñas de IBM MQ . Para obtener más información sobre los métodos de protección de la contraseña del repositorio de claves, consulte "Cifrado de contraseñas de repositorio de claves en IBM i" en la página 282.

Los campos KeyRepoPassword de MQSCO

Para proporcionar una contraseña de repositorio de claves utilizando la estructura MQSCO, debe utilizar una combinación de los tres campos de serie de variables siguientes:

KeyRepoPasswordLength
La longitud de la contraseña.

KeyRepoPasswordPtr
Puntero a la ubicación en la memoria que contiene la contraseña.

KeyRepoPasswordOffset
La ubicación de la contraseña en la memoria, representada como número de bytes desde el inicio de la estructura MQSCO.

Nota: Sólo puede proporcionar uno de **KeyRepoPasswordPtr** o **KeyRepoPasswordOffset**.

Por ejemplo:

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

Atención: Si proporciona la contraseña utilizando este método, cifre la contraseña antes de que se proporcione a la aplicación IBM MQ client . Para obtener más información, consulte "Cifrado de la contraseña del repositorio de claves" en la página 288.

Para obtener más información sobre la estructura MQCSO, consulte MQSCO-Opciones de configuración SSL/TLS.

La variable de entorno *MQKEYRPWD*

Si no se proporciona una contraseña de repositorio de claves al cliente utilizando la estructura MQSCO, puede especificar la contraseña de repositorio de claves utilizando la variable de entorno *MQKEYRPWD* . Por ejemplo:

```
export MQKEYRPWD=passw0rd
```

o

```
set MQKEYRPWD=passw0rd
```

donde *passw0rd* es la contraseña.

Atención: Si proporciona la contraseña utilizando este método, cifre la contraseña antes de establecer el valor de la variable de entorno. Para obtener más información, consulte “Cifrado de la contraseña del repositorio de claves” en la página 288.

El atributo SSLKeyRepositoryPassword del archivo de configuración del cliente

Si no se proporciona una contraseña del repositorio de claves al cliente utilizando uno de los otros métodos, puede especificar la contraseña del repositorio de claves utilizando el atributo **SSLKeyRepositoryPassword** en la stanza **SSL** del archivo de configuración del cliente. Por ejemplo:

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

Atención: Si proporciona la contraseña utilizando este método, cifre la contraseña antes de establecer el valor del atributo **SSLKeyRepositoryPassword** . Para obtener más información, consulte “Cifrado de la contraseña del repositorio de claves” en la página 288.

Para obtener más información sobre la stanza SSL del archivo de configuración de cliente, consulte Stanza SSL del archivo de configuración de cliente.

El archivo de ocultación del repositorio de claves

Si la contraseña del repositorio de claves no se proporciona al cliente utilizando uno de los otros métodos, IBM MQ presupone que existe un archivo de ocultación en el mismo directorio que el repositorio de claves. El archivo stash tiene el mismo nombre de raíz que el repositorio de claves, pero tiene la extensión `.sth` .

Se crea un archivo de ocultación de repositorio de claves utilizando la herramienta de línea de mandatos **amqrsslc** . Para crear el archivo stash, ejecute el mandato siguiente:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

Este mandato le solicita la contraseña que debe cifrar. La contraseña se cifra mediante el sistema de protección de contraseñas de IBM MQ , con una clave de cifrado predeterminada a menos que se proporcione una utilizando el parámetro **-sf** .

Para obtener más información, consulte “Programa de utilidad de Cliente SSL de IBM MQ (amqrsslc) para IBM i” en la página 297 y “Cifrado de la contraseña del repositorio de claves” en la página 288.

Cifrado de la contraseña del repositorio de claves

Si proporciona la contraseña del repositorio de claves utilizando cualquier método que no sea un archivo de ocultación, cifre la contraseña utilizando el sistema de protección de contraseñas de IBM MQ . Para cifrar la contraseña, ejecute el mandato **runmqicred** . Especifique la contraseña del repositorio de claves cuando se le solicite. El mandato genera la contraseña cifrada. La contraseña cifrada se puede proporcionar a IBM MQ MQI client en lugar de a la contraseña de texto sin formato utilizando cualquiera de los métodos descritos.

Una clave de cifrado, que se conoce como la clave inicial, se utiliza para cifrar la contraseña. Cuando cifre la contraseña, utilice una clave inicial exclusiva para proteger de forma segura la contraseña. Para proporcionar su propia clave inicial, utilice el parámetro **-sf** para el mandato **runmqicred** . Si no proporciona una clave inicial, se utiliza la clave predeterminada.

Para obtener más información, consulte runmqicred (proteger contraseñas de cliente de IBM MQ).

Si proporciona su propia clave inicial cuando la contraseña del repositorio de claves está cifrada y proporciona la contraseña cifrada al IBM MQ MQI client, también debe asegurarse de que proporciona la misma clave inicial al IBM MQ MQI client. Para obtener más información sobre cómo proporcionar la clave inicial a un IBM MQ MQI client, consulte "Suministro de una clave inicial para un IBM MQ MQI client en IBM i" en la página 289.

Conceptos relacionados

"Cifrado de contraseñas de repositorio de claves en IBM i" en la página 282
Varios componentes de IBM MQ necesitan acceso a un repositorio de claves que contenga certificados digitales o claves simétricas. Un repositorio de claves está protegido con una contraseña, ya que contiene información confidencial. La contraseña del repositorio de claves debe almacenarse en una ubicación en la que IBM MQ pueda leerla cuando se acceda al repositorio de claves. La contraseña también debe estar cifrada para reducir la probabilidad de acceso no autorizado al repositorio de claves.

"Suministro de la contraseña del repositorio de claves para un gestor de colas en IBM i" en la página 286
Puesto que el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

IBM i *Suministro de una clave inicial para un IBM MQ MQI client en IBM i*

Si proporciona variables a un IBM MQ MQI client que se ha cifrado utilizando el sistema de protección por contraseña de IBM MQ , es posible que tenga que proporcionar la clave inicial correspondiente que se ha utilizado para cifrar el valor.

Si no ha especificado una clave inicial al cifrar el valor, no es necesario que proporcione ningún valor de clave inicial a IBM MQ client. Sin embargo, si ha utilizado una clave inicial exclusiva, puede proporcionar la clave inicial a IBM MQ client utilizando los métodos siguientes:

- "Suministro de la clave inicial utilizando la estructura MQCSP" en la página 289
- "Suministro de la clave inicial utilizando la variable de entorno MQS_MQI_KEYFILE" en la página 290
- "Suministro de la clave inicial utilizando el archivo de configuración de cliente" en la página 290

Suministro de la clave inicial utilizando la estructura MQCSP

Para proporcionar la clave inicial utilizando la estructura MQCSP, debe utilizar una combinación de los tres campos de serie de variables siguientes:

InitialKeyLength
La longitud de la clave inicial

InitialKeyPtr
Un puntero a la ubicación en la memoria que contiene la clave inicial

InitialKeyOffset
La ubicación de la clave inicial en la memoria, representada como número de bytes desde el inicio de la estructura MQCSP.

Nota: Sólo puede proporcionar uno de **InitialKeyPtr** o **InitialKeyOffset**.

Por ejemplo:

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
```

```
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

Suministro de la clave inicial utilizando la variable de entorno MQS_MQI_KEYFILE

Si no se proporciona una clave inicial al cliente utilizando la estructura MQCSP, IBM MQ comprueba la variable de entorno *MQS_MQI_KEYFILE* . Debe establecer esta variable de entorno en la ubicación de un archivo que contenga una sola línea de texto, que conste de la clave inicial que desea utilizar.

Por ejemplo, si existe un archivo denominado `mykey.key` en el directorio raíz y contiene la clave inicial, debe establecer la variable de entorno como se indica a continuación:

```
export MQS_MQI_KEYFILE=/mykey.key
```

o

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

Suministro de la clave inicial utilizando el archivo de configuración de cliente

Si no se proporciona una clave inicial al cliente utilizando un mecanismo anterior, IBM MQ comprueba el atributo **MQIInitialKeyFile** de la stanza Security del archivo `mqclient.ini` . Debe establecer este atributo en la ubicación de un archivo que contenga una sola línea de texto, que conste de la clave inicial que desea utilizar.

Por ejemplo, si existe un archivo denominado `mykey.key` en el directorio raíz y contiene la clave inicial, el archivo de configuración del cliente debe contener lo siguiente:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

Conceptos relacionados

“Cifrado de contraseñas de repositorio de claves en IBM i” en la página 282
Varios componentes de IBM MQ necesitan acceso a un repositorio de claves que contenga certificados digitales o claves simétricas. Un repositorio de claves está protegido con una contraseña, ya que contiene información confidencial. La contraseña del repositorio de claves debe almacenarse en una ubicación en la que IBM MQ pueda leerla cuando se acceda al repositorio de claves. La contraseña también debe estar cifrada para reducir la probabilidad de acceso no autorizado al repositorio de claves.

“Trabajar con SSL/TLS en IBM i” en la página 281
Esta colección de temas ofrece instrucciones de tareas individuales relacionadas con TLS (Seguridad de la capa de transporte) en IBM MQ for IBM i.

Crear una entidad emisora de certificados y un certificado para pruebas en IBM i

Utilice este procedimiento para crear un certificado de CA local para firmar peticiones de certificado, y para crear e instalar el certificado de CA.

Antes de empezar

Las instrucciones de este tema presuponen que no existe una autorización de certificado (CA) local. Si no existe una CA local, vaya a “Solicitar un certificado de servidor en IBM i” en la página 291.

Acerca de esta tarea

Los certificados de CA que se proporcionan cuando se instala TLS están firmados por la CA emisora. En IBM i, puede generar una entidad emisora de certificados local que pueda firmar certificados de servidor para probar las comunicaciones TLS en el sistema. Siga estos pasos en un navegador web para crear un certificado de CA local:

Procedimiento

1. Acceda a la interfaz DCM, tal y como se describe en el apartado "Acceso a DCM" en la página 281.
2. En el panel de navegación, pulse **Crear una Entidad emisora de certificados**.
   La página Crear una Entidad emisora de certificados se visualiza en la sección de tareas.
3. Escriba una contraseña en el campo **Contraseña del almacén de certificados** y vuelva a escribirla en el campo **Confirmar contraseña**.
4. Escriba un nombre en el campo **Nombre de entidad emisora de certificados (CA)** , por ejemplo TLS `Test Certificate Authority`.
5. Escriba valores adecuados en los campos **Nombre común** y **Organización** y seleccione un país. En cuanto a los campos opcionales restantes, escriba los valores que necesite.
6. Escriba un periodo de validez para la CA local en el campo **Periodo de validez**.
   El valor predeterminado es 1095 días.
7. Pulse **Continuar**.
   Se crea la CA y DCM crea un almacén de certificados y un certificado de CA para la CA local.
8. Pulse **Instalar certificado**.
   Se visualiza el recuadro de diálogo del gestor de descargas.
9. Escriba el nombre de vía de acceso completo del archivo temporal en el que desea almacenar el certificado de CA y pulse **Guardar**.
10. Cuando el proceso de descarga haya finalizado, pulse **Abrir**.
    Se visualiza la ventana Certificado.
11. Pulse **Instalar certificado**.
    Se visualiza el asistente Importar certificado.
12. Pulse **Siguiente**.
13. Seleccione **Seleccionar automáticamente el almacén de certificados basándose en el tipo de certificado** y pulse **Siguiente**.
14. Pulse **Finalizar**.
    Se visualiza una ventana de confirmación.
15. Pulse **Aceptar**.
16. En la ventana Certificado, pulse **Aceptar**.
17. Pulse **Continuar**.
    Se visualiza la página Política de Entidad emisora de certificados en la sección de tareas.
18. En el campo **Permitir la creación de certificados de usuario**, seleccione **Sí**.
19. En el campo **Periodo de validez**, escriba el periodo de validez de los certificados emitidos por la CA local.
    El valor predeterminado es 365 días.
20. Pulse **Continuar**.
    Se visualiza la página Crear un certificado en almacén de certificados nuevo en la sección de tareas.
21. Compruebe que ninguna de las aplicaciones está seleccionada.
22. Pulse **Continuar** para completar la configuración de la CA local.

Qué hacer a continuación

Si necesita renovar un certificado existente, consulte Renovación de un certificado existente en la documentación de IBM i .

Solicitar un certificado de servidor en IBM i

Los certificados digitales protegen contra la suplantación de identidad, certificando que una clave pública pertenece a una entidad especificada. Se puede solicitar un nuevo certificado de servidor de una entidad emisora de certificados utilizando el Gestor de certificados digitales (DCM).

Acerca de esta tarea

Realice los pasos siguientes en un navegador web:

Procedimiento

1. Acceda a la interfaz DCM, tal y como se describe en el apartado “Acceso a DCM” en la página 281.
2. En el panel de navegación, pulse **Seleccionar un almacén de certificados**.

   La página Seleccionar un almacén de certificados se visualiza en la sección de tareas.
3. Seleccione el almacén de certificados que desea utilizar y pulse **Continuar**.
4. Opcional: Si ha seleccionado ***SYSTEM** en el paso 3, entre la contraseña del almacén del sistema y pulse **Continuar**.
5. Opcional: Si ha seleccionado **Otro almacén de certificados del sistema** en el paso 3, en el campo **Vía de acceso y nombre de archivo del almacén de certificados**, escriba la vía de acceso y nombre de archivo de IFS que estableció cuando creó el almacén de certificados. Además, escriba una contraseña en el campo **Contraseña del almacén de certificados**. A continuación, pulse **Continuar**
6. En el panel de navegación, pulse **Crear certificado**.
7. En la sección de tareas, seleccione el botón **Certificado de servidor o de cliente** y pulse **Continuar**.

   La página Seleccionar una Entidad emisora de certificados (CA) se visualiza en la sección de tareas.
8. Si tiene una CA local en la estación de trabajo, elija la CA local o una CA comercial para firmar el certificado. Seleccione el botón de selección correspondiente a la CA que desee y pulse **Continuar**.

   La página Crear un certificado se visualiza en la sección de tareas.
9. Opcional: Para un gestor de colas, en el campo **Etiqueta de certificado**, escriba la etiqueta del certificado.

   La etiqueta es el valor del atributo **CERTLABL**, si éste está establecido o el valor predeterminado `ibmwebspheremq` con el nombre del gestor de colas añadido, todo en minúsculas. Para obtener información detallada, consulte Etiquetas de certificados digitales.

   Por ejemplo, para el gestor de colas QM1, escriba `ibmwebspheremqqm1` para utilizar el valor predeterminado.
10. Opcional: Para un IBM MQ MQI client, en el campo **Etiqueta de certificado**, escriba `ibmwebspheremq` seguido del ID de usuario de inicio de sesión en minúsculas.
    Por ejemplo, escriba `ibmwebspheremqmyuserID`
11. Escriba valores adecuados en los campos **Nombre común** y **Organización** y seleccione un país. En cuanto a los campos opcionales restantes, escriba los valores que necesite.

Resultados

Si ha seleccionado una CA comercial para firmar el certificado, DCM crea una solicitud de certificado en formato PEM (Privacy-Enhanced Mail). Reenvíe la solicitud a la CA que haya elegido.

Si ha seleccionado la CA local para firmar el certificado, DCM le informa de que el certificado se ha creado en el almacén de certificados y que se puede utilizar.

Solicitud de un certificado de servidor para un sistema remoto en IBM i

Siga este procedimiento para crear un certificado firmado por la entidad emisora de certificados (CA) local, o para solicitar un certificado de servidor firmado por una CA comercial para importarlo a un repositorio de claves en otras plataformas.

Acerca de esta tarea

Debe utilizarse un certificado de usuario cuando el Gestor de certificados digitales (DCM) actúa como gestor de certificados para IBM MQ en varias plataformas. Para los certificados personales que se distribuyen a otras plataformas y se importan en un repositorio de claves, realice los pasos siguientes en un navegador web:

Procedimiento

1. Acceda a la interfaz DCM, tal y como se describe en el apartado "Acceso a DCM" en la página 281.
2. En el panel de **navegación**, pulse **Crear certificado**.

   La página **Crear certificado** se visualiza en la sección de tareas.
3. En el panel **Crear certificado**, seleccione el botón **Certificado de usuario** y pulse **Continuar**.

   Se visualiza la página **Crear certificado de usuario**.
4. En el panel **Crear certificado de usuario**, rellene los campos obligatorios bajo Información del certificado para **Nombre de organización**, **Estado** o **provincia**, **País** o **región**. Opcionalmente, entre valores en los campos **Unidad de organización** y **Localidad** o **ciudad**. Pulse **Continuar**.

   El **Nombre común** se establece automáticamente en el ID de usuario con el que ha iniciado la sesión en el sistema iSeries.
5. En el siguiente panel **Crear certificado de usuario**, pulse **Instalar certificado** y pulse **Continuar**.

   Se visualiza un mensaje que indica que `Se ha instalado su certificado personal. Debería guardar una copia de seguridad de este certificado.`
6. Pulse **Aceptar**.
7. En función del navegador web que haya utilizado para acceder a DCM, realice uno de los pasos siguientes:
   - Para Microsoft Edge, elija: **Herramientas > Opciones de Internet > separador Contenido > botón Certificados > separador Personal >**. Seleccione el certificado y pulse **Exportar**.
   - Para Mozilla Firefox , seleccione: **Herramientas > Opciones > Avanzado > Pestaña Cifrado > botón Ver certificados > pestaña Sus certificados >**. Seleccione el certificado y pulse **Copia de seguridad**. Seleccione la vía de acceso y el nombre de archivo y pulse **Aceptar**.
8. Transfiera el certificado exportado al sistema remoto utilizando FTP en formato binario.
9. Importe el certificado que se ha exportado en el paso "7" en la página 293 al repositorio de claves en el sistema remoto.
   - Si el certificado se ha guardado utilizando Microsoft Edge, utilice las instrucciones descritas en el archivo "Importación de un certificado personal desde un archivo Microsoft.pfx" en la página 565 .
   - Si el certificado se ha guardado utilizando Mozilla Firefox, siga las instrucciones que se indican en Importación de un certificado personal a un repositorio de claves.

   Durante la importación, asegúrese de que el nombre de etiqueta del certificado personal y el certificado de firmante se han cambiado al valor que IBM MQ espera. La etiqueta debe ser el valor del atributo IBM MQ gestor de colas **CERTLABL** , si está establecido, o el valor predeterminado de `ibmwebspheremq` con el nombre del gestor de colas añadido, todo en minúsculas. Para obtener más información, consulte Etiquetas de certificado digital.

Añadir certificados de servidor a un repositorio de claves en IBM i

Siga este procedimiento para añadir un certificado solicitado al repositorio de claves.

Acerca de esta tarea

Después de que la CA envíe un nuevo certificado de servidor, debe añadirlo al almacén de certificados desde el que ha generado la solicitud. Si la CA envía el certificado como parte de un mensaje de correo electrónico, copie el certificado en un archivo aparte.

Nota:

- No necesita efectuar este procedimiento si la CA local firma el certificado de servidor.
- Antes de importar un certificado de servidor con el formato PKCS #12 a DCM, deberá importar primero el certificado de CA correspondiente.

Utilice el siguiente procedimiento para recibir un certificado de servidor en el almacén de certificados del gestor de colas:

Procedimiento

1. Acceda a la interfaz DCM, tal y como se describe en el apartado "Acceso a DCM" en la página 281.
2. En la categoría de tareas **Gestionar certificados** del panel de navegación, pulse **Importar certificado**.

   Se visualiza la página Importar certificado en la sección de tareas.
3. Seleccione el botón de selección correspondiente al tipo de certificado y pulse **Continuar**.

   Se visualiza la página Importar certificado de servidor o de cliente o Importar certificado de Entidad emisora de certificados (CA) en la sección de tareas.
4. En el campo **Importar archivo**, escriba el nombre de archivo del certificado que desea importar y pulse **Continuar**.

   DCM determina automáticamente el formato del archivo.
5. Si el certificado es un certificado de **Servidor o cliente**, escriba la contraseña en la sección de tareas y pulse **Continuar**.

   DCM le informa de que se ha importado el certificado.

Exportar un certificado desde un repositorio de claves en IBM i

Exportar un certificado exporta ambas claves, la pública y privada. Esta acción se deberá realizar con extrema cautela, ya que pasar una clave privada comprometería por completo la seguridad.

Antes de empezar

Cuando se comparte un certificado de usuario con otro usuario, se intercambian claves públicas. Este proceso se describe en la tarea 5 de **. Compartir certificados** en la sección Compartir certificados de "Guía de inicio rápido para AMS en AIX and Linux" en la página 625. Al exportar un certificado tal como se describe aquí, se exportan ambas claves, pública y privada. Esta acción se deberá realizar con extrema cautela, ya que pasar una clave privada comprometería por completo la seguridad.

Acerca de esta tarea

Realice los pasos siguientes en el sistema desde el que desea exportar el certificado:

Procedimiento

1. Acceda a la interfaz DCM, tal y como se describe en el apartado "Acceso a DCM" en la página 281.
2. En el panel de navegación, pulse **Seleccionar un almacén de certificados**.

   La página Seleccionar un almacén de certificados se visualiza en la sección de tareas.
3. Seleccione el almacén de certificados que desea utilizar y pulse **Continuar**.
4. Opcional: Si ha seleccionado ***SYSTEM** en el paso 3, entre la contraseña del almacén del sistema y pulse **Continuar**.
5. Opcional: Si ha seleccionado **Otro almacén de certificados del sistema** en el paso 3, en el campo **Vía de acceso y nombre de archivo del almacén de certificados**, escriba la vía de acceso y nombre de archivo de IFS que estableció cuando creó el almacén de certificados y escriba una contraseña en el campo **Contraseña del almacén de certificados**. A continuación, pulse **Continuar**
6. En la categoría de tareas **Gestionar certificados** del panel de navegación, pulse **Exportar certificado**.

   La página Exportar un certificado se visualiza en la sección de tareas.
7. Seleccione el botón de selección correspondiente al tipo de certificado y pulse **Continuar**.

   Se visualiza la página Exportar certificado de servidor o de cliente o Exportar certificado de Entidad emisora de certificados (CA) en la sección de tareas.
8. Seleccione el certificado que desea exportar.
9. Seleccione el botón de selección para especificar si desea exportar el certificado a un archivo o directamente a otro almacén de certificados.

10. Si ha seleccionado exportar un certificado de servidor o de cliente a un archivo, facilite la siguiente información:
    - La vía de acceso y nombre de archivo de la ubicación donde desea almacenar el certificado exportado.
    - Para un certificado personal, la contraseña que se utiliza para cifrar el certificado exportado y el release de destino. En el caso de certificados de CA, no necesita especificar la contraseña.
11. Si ha seleccionado exportar un certificado directamente a otro almacén de certificados, especifique el almacén de certificados de destino y la contraseña.
12. Pulse **Continuar**.

Importar un certificado a un repositorio de claves en IBM i

Siga este procedimiento para importar un certificado.

Antes de empezar

Antes de importar un certificado personal con el formato PKCS #12 a DCM, deberá importar primero el certificado de CA correspondiente.

Acerca de esta tarea

Realice estos pasos en la máquina a la que desea importar el certificado.

Procedimiento

1. Acceda a la interfaz DCM, tal y como se describe en el apartado "Acceso a DCM" en la página 281.
2. En el panel de navegación, pulse **Seleccionar un almacén de certificados**.

   La página Seleccionar un almacén de certificados se visualiza en la sección de tareas.
3. Seleccione el almacén de certificados que desea utilizar y pulse **Continuar**.
4. Opcional: Si ha seleccionado ***SYSTEM** en el paso 3, entre la contraseña del almacén del sistema y pulse **Continuar**.
5. Opcional: Si ha seleccionado **Otro almacén de certificados del sistema** en el paso 3, en el campo **Vía de acceso y nombre de archivo del almacén de certificados**, escriba la vía de acceso y nombre de archivo de IFS que estableció cuando creó el almacén de certificados y escriba una contraseña en el campo **Contraseña del almacén de certificados**. A continuación, pulse **Continuar**
6. En la categoría de tareas **Gestionar certificados** del panel de navegación, pulse **Importar certificado**.

   La página Importar certificado se visualiza en la sección de tareas.
7. Seleccione el botón de selección correspondiente al tipo de certificado y pulse **Continuar**.

   Se visualiza la página Importar certificado de servidor o de cliente o Importar certificado de Entidad emisora de certificados (CA) en la sección de tareas.
8. En el campo **Importar archivo**, escriba el nombre de archivo del certificado que desea importar y pulse **Continuar**.

   DCM determina automáticamente el formato del archivo.
9. Si el certificado es un certificado de **Servidor o cliente**, escriba la contraseña en la sección de tareas y pulse **Continuar**. DCM le informa de que se ha importado el certificado.

Suprimir certificados en IBM i

Utilice este procedimiento para suprimir certificados personales.

Procedimiento

1. Acceda a la interfaz DCM, tal y como se describe en el apartado "Acceso a DCM" en la página 281.
2. En el panel de navegación, pulse **Seleccionar un almacén de certificados**.

   La página Seleccionar un almacén de certificados se visualiza en la sección de tareas.
3. Seleccione el recuadro **Otro almacén de certificados del sistema** y pulse **Continuar**.

Se visualiza la página Almacén de certificados y contraseña.

4. En el campo **Vía de acceso y nombre de archivo del almacén de certificados**, escriba la vía de acceso y nombre de archivo de IFS que estableció cuando creó el almacén de certificados.
5. Escriba una contraseña en el campo **Contraseña del almacén de certificados**. Pulse **Continuar**.
   La página Almacén de certificados actual se visualiza en la sección de tareas.
6. En la categoría de tareas **Gestionar certificados** del panel de navegación, pulse **Suprimir certificado**.
   La página Confirmar supresión de certificado se visualiza en la sección de tareas.
7. Seleccione el certificado que desea suprimir. Pulse **Suprimir**.
8. Pulse **Sí** para confirmar que desea suprimir el certificado. De lo contrario, pulse **No**.
   DCM le informa si ha suprimido el certificado.

*Utilización del almacén de certificados *SYSTEM para la autenticación unidireccional en IBM i*

Siga estas instrucciones para configurar la autenticación unidireccional.

Antes de empezar

- Cree un gestor de colas, canales y colas de transmisión.
- Cree un certificado de servidor o de cliente en el gestor de colas del servidor.
- Transfiera el certificado de CA al gestor de colas del cliente e impórtelo al repositorio de claves.
- Inicie un escucha en los gestores de colas del servidor y del cliente.

Acerca de esta tarea

Para utilizar la autenticación unidireccional, utilizando un sistema que ejecuta IBM i como servidor TLS, establezca el parámetro Repositorio de claves SSL (SSLKEYR) en *SYSTEM. Este valor registra el gestor de colas de IBM MQ como una aplicación. Podrá entonces asignar un certificado al gestor de colas para permitir la autenticación unidireccional.

También puede utilizar almacenes de claves privados para implementar la autenticación unidireccional creando un certificado ficticio para el gestor de colas del cliente en el depósito de claves.

Procedimiento

1. Realice los pasos siguientes en los gestores de colas del servidor y del cliente:
   a) Modifique el gestor de colas para establecer el parámetro SSLKEYR, emitiendo el mandato `CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)`.
   b) Oculte la contraseña para el depósito de claves predeterminado, emitiendo el mandato `CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')`.
      La contraseña debe estar entre comillas simples.
   c) Modifique los canales para que tengan la CipherSpec correcta en el parámetro SSLCIPHER.
   d) Renueve la seguridad TLS emitiendo el mandato `RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)`.
2. Asigne el certificado al gestor de colas del servidor utilizando DCM, como se indica a continuación:
   a) Acceda a la interfaz DCM, tal y como se describe en el apartado "Acceso a DCM" en la página 281.
   b) En el panel de navegación, pulse **Seleccionar un almacén de certificados**.
      La página Seleccionar un almacén de certificados se visualiza en la sección de tareas.
   c) Seleccione el almacén de certificados *SYSTEM y pulse **Continuar**.
   d) En el panel de la izquierda, expanda **Gestionar aplicaciones**.
   e) Seleccione **Ver definición de aplicación** para comprobar que el gestor de colas se ha registrado como una aplicación.
      `SSL (WMQ)` aparece listado en la tabla.

f) Seleccione **Actualizar asignación de certificados**.

g) Seleccione **Servidor** y pulse **Continuar**.

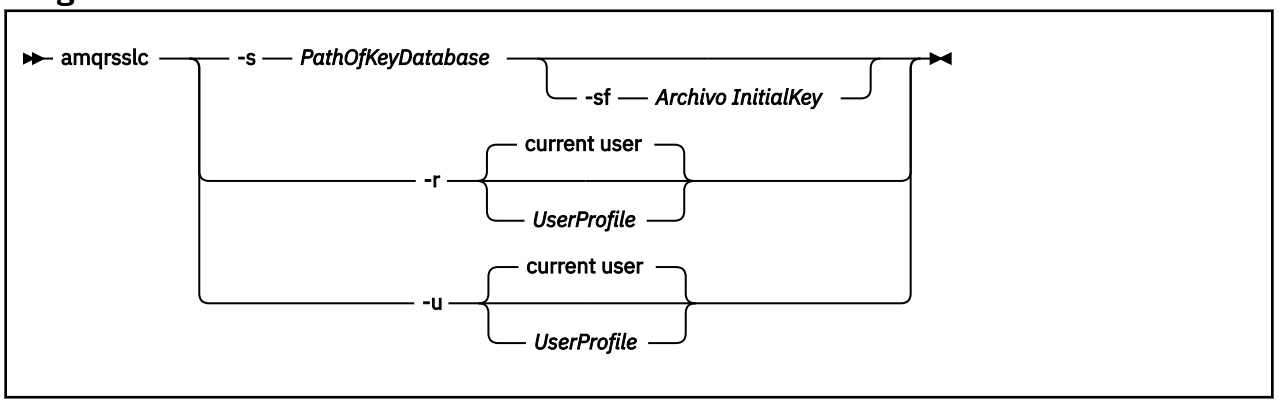
h) Seleccione QMGRNAME (WMQ) y pulse **Actualizar asignación de certificados**.

i) Seleccione el certificado y pulse **Asignar nuevo certificado**. Se abre una ventana que le informa de que el certificado se ha asignado a la aplicación.

Programa de utilidad de Cliente SSL de IBM MQ (amqrsslc) para IBM i

El programa de utilidad de Cliente SSL de IBM MQ (amqrsslc) para IBM i lo utiliza el IBM MQ MQI client En los sistemas IBM i para registrar o anular el registro del perfil de usuario del cliente o para ocultar la contraseña del almacén de certificados. El programa de utilidad sólo lo puede ejecutar un usuario con un perfil con autorización especial *ALLOBJ o un miembro del grupo QMQMADM que tenga opciones para crear o suprimir registros de aplicación en el Gestor de certificados digitales (DCM).

Diagrama de sintaxis



Registrar el perfil de usuario del cliente

Si IBM MQ MQI client utiliza el almacén de certificados *SYSTEM, debe registrar el perfil de usuario de cliente (usuario de inicio de sesión) para utilizarlo como una aplicación con Digital Certificate Manager (DCM).

Si desea registrar el perfil de usuario del cliente, ejecute el programa **amqrsslc** con la opción `-r` con *PerfilUsuario*. El perfil de usuario utilizado al llamar a **amqrsslc** debe tener autorización *USE. Al especificar el *PerfilUsuario* con la opción `-r`, se registra el *PerfilUsuario* como una aplicación de servidor con una etiqueta de aplicación exclusiva de QIBM_WEBSPHERE_MQ_*PerfilUsuario* y una etiqueta con una descripción de *PerfilUsuario* (WMQ). Esta aplicación de servidor se mostrará entonces en el DCM y podrá asignar a esta aplicación cualquier certificado de servidor o de cliente del almacén del sistema.

Nota: Si un perfil de usuario no se especifica con la opción `-r`, se registra el perfil de usuario del usuario que ejecuta la herramienta **amqrsslc**.

El código siguiente utiliza **amqrsslc** para registrar un perfil de usuario. En el primer ejemplo, se registra el perfil de usuario especificado; en el segundo, es el perfil del usuario que ha iniciado la sesión:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

Anular el registro del perfil de usuario del cliente

Para anular el registro del perfil del cliente, ejecute el programa **amqrsslc** con la opción `-u` con *PerfilUsuario*. El perfil de usuario utilizado al llamar a **amqrsslc** debe tener autorización *USE. Al proporcionar al *PerfilUsuario* la etiqueta `-u` se anula el registro de *PerfilUsuario* con la etiqueta QIBM_WEBSPHERE_MQ_*PerfilUsuario* del DCM.

Nota: Si un perfil de usuario no se especifica con la opción –u, se anula el registro del perfil de usuario del usuario que ejecuta la herramienta **amqrsslc**.

El código siguiente utiliza **amqrsslc** para anular el registro de un perfil de usuario. En el primer ejemplo, se anula el registro del perfil de usuario especificado; en el segundo, es el perfil del usuario que ha iniciado la sesión:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

Ocultar la contraseña del almacén de certificados

Si IBM MQ MQI client no utiliza el almacén de certificados *SYSTEM y utiliza otro almacén de certificados (es decir, MQSSLKEYR se establece en un valor distinto de *SYSTEM), la contraseña de la base de datos de claves puede ocultarse para que la aplicación cliente no la tenga que especificar cuando se ejecute.

Utilice la opción `-s` para ocultar la contraseña de la base de datos de claves. Especifique la vía de acceso completa y el nombre de la base de datos de claves. Si no se proporciona la extensión de archivo, se presupone que es `.kdb`.

En el código siguiente, el nombre de archivo completo del almacén de certificados es `/Path/Of/KeyDatabase/MyKey.kdb`:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

La ejecución de este código da como resultado una solicitud de la contraseña de esta base de datos de claves. Esta contraseña se oculta en un archivo con el mismo nombre que la base de datos de claves con una extensión `.sth` .

Además, se puede especificar la clave inicial para cifrar la contraseña. La clave inicial debe almacenarse en un archivo como una sola línea de texto y, a continuación, la ubicación de dicho archivo se proporciona al programa a través del distintivo **-sf** . Si no se proporciona ningún archivo de claves inicial, se utiliza una clave predeterminada para cifrar la contraseña.

El archivo de ocultación se almacena en la misma vía de acceso que la base de datos de claves. El ejemplo de código genera un archivo de ocultación `/Path/Of/KeyDatabase/MyKey.sth`.

QMQM es el propietario de usuario y QMQMADM el propietario del grupo para este archivo. QMQM y QMQMADM tienen permiso de lectura y grabación, y otros perfiles sólo tienen permiso de lectura.

Cuándo entran en vigor los cambios en los certificados o en el almacén de certificados en IBM i

Cuando cambia los certificados de un almacén de certificados, o la ubicación del almacén de certificados, los cambios entran en vigor dependiendo del tipo de canal y de cómo se ejecuta el canal.

Los cambios efectuados en los certificados del almacén de certificados y en el atributo de repositorio de claves entran en vigor en las siguientes situaciones:

- Cuando un nuevo proceso de canal de salida individual ejecuta por primera vez un canal TLS.
- Cuando un nuevo proceso de canal de entrada individual TCP/IP recibe por primera vez una solicitud para iniciar un canal TLS.
- Cuando se emite el mandato MQSC REFRESH SECURITY TYPE(SSL) para renovar el entorno TLS de IBM MQ.
- Para los procesos de la aplicación cliente, cuando se cierra la última conexión TLS del proceso. La siguiente conexión TLS captará los cambios del certificado.
- Para canales que se ejecutan como hebras de un proceso de agrupación de procesos (amqrmppa), cuando se inicia o se reinicia el proceso de agrupación de procesos y ejecuta por primera vez un canal TLS. Si el proceso de agrupación de procesos ya ha ejecutado un canal TLS y desea que el cambio entre en vigor de forma inmediata, ejecute el mandato MQSC REFRESH SECURITY TYPE(SSL).

- Para canales que se ejecutan como hebras del iniciador de canal, cuando se inicia o se reinicia el iniciador de canal y ejecuta por primera vez un canal TLS. Si el proceso iniciador de canal ya ha ejecutado un canal TLS y desea que el cambio entre en vigor de forma inmediata, ejecute el mandato MQSC REFRESH SECURITY TYPE(SSL).
- Para canales que se ejecutan como hebras de un escucha TCP/IP, cuando se inicia o se reinicia el escucha y recibe por primera vez una solicitud para iniciar un canal TLS. Si el escucha ya ha ejecutado un canal TLS y desea que el cambio entre en vigor de forma inmediata, ejecute el mandato MQSC REFRESH SECURITY TYPE(SSL).

Configurar el hardware de cifrado en IBM i

Utilice este procedimiento para configurar el Coprocesador criptográfico en IBM i

Antes de empezar

Asegúrese de que el perfil de usuario tenga las autorizaciones especiales *ALLOBJ y *SECADM para configurar el hardware del coprocesador.

Procedimiento

1. Vaya a `http://`*`machine`*`.domain:2001` o `https://`*`machine`*`.domain:2010`, donde *máquina* es el nombre del sistema.

   Se visualiza un recuadro de diálogo que le solicita un nombre de usuario y una contraseña.
2. Escriba un perfil de usuario y una contraseña válidos de IBM i.
3. Vaya a Criptografía y siga los correspondientes enlaces para obtener más información.

Qué hacer a continuación

Para obtener información más concreta sobre la configuración del Coprocesador criptográfico 4767, consulte Coprocesador criptográfico 4767.

ALW Trabajar con SSL/TLS en AIX, Linux, and Windows

En sistemas AIX, Linux, and Windows, el soporte TLS (Transport Layer Security) se instala con IBM MQ.

Nota: V 9.4.0 Deprecated A partir de IBM MQ 9.4.0, el uso de repositorios de claves CMS y archivos de ocultación con aplicaciones IBM MQ Java está en desuso. Migre a la utilización de repositorios de claves PKCS #12 y proteja las contraseñas de repositorio de claves utilizando el sistema de protección de contraseñas de IBM MQ .

Importante: V 9.4.0 V 9.4.0 A partir de IBM MQ 9.4.0, los repositorios de claves CMS y los archivos de ocultación no están soportados con los canales AMQP y MQTT que utilizan SSL/TLS. Utilice los repositorios de claves PKCS #12 y proteja las contraseñas del repositorio de claves utilizando en su lugar el sistema de protección de contraseñas IBM MQ .

Para obtener más información sobre las políticas de validación de certificados, consulte Validación de certificados y diseño de políticas de confianza.

Para obtener más información sobre los mandatos que se utilizan para gestionar repositorios de claves y certificados en AIX, Linux, and Windows, consulte “Mandatos runmqakm y runmqktool en AIX, Linux, and Windows” en la página 551.

ALW *Configuración de un repositorio de claves en AIX, Linux, and Windows*

Siga este procedimiento para crear un nuevo repositorio de claves.

Antes de empezar

Un repositorio de claves está protegido con una contraseña, ya que contiene información confidencial. Antes de crear el repositorio de claves, revise las opciones que proporciona IBM MQ para almacenar de

forma segura la contraseña del repositorio de claves. Para obtener más información, consulte “Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows” en la página 302.

Nota: V 9.4.0 Deprecated A partir de IBM MQ 9.4.0, el uso de repositorios de claves CMS y archivos de ocultación con aplicaciones IBM MQ Java está en desuso. Migre a la utilización de repositorios de claves PKCS #12 y proteja las contraseñas de repositorio de claves utilizando el sistema de protección de contraseñas de IBM MQ .

Importante: V 9.4.0 V 9.4.0 A partir de IBM MQ 9.4.0, los repositorios de claves CMS y los archivos de ocultación no están soportados con los canales AMQP y MQTT que utilizan SSL/TLS. Utilice los repositorios de claves PKCS #12 y proteja las contraseñas del repositorio de claves utilizando en su lugar el sistema de protección de contraseñas IBM MQ . Puede crear un repositorio de claves PKCS #12 utilizando el mandato siguiente:

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

Este mandato crea un archivo de repositorio de claves PKCS #12 denominado *filename*.p12 que está protegido con la contraseña especificada.

Acerca de esta tarea

Una conexión TLS requiere un *depósito de claves* en cada extremo de la conexión. Cada gestor de colas de IBM MQ y IBM MQ MQI client debe tener acceso a un repositorio de claves. Para obtener más información, consulte “Repositorio de claves SSL/TLS” en la página 26.

Los certificados digitales se almacenan en el repositorio de claves. Estos certificados digitales tienen etiquetas. La etiqueta de certificado asocia un certificado personal con un gestor de colas específico o IBM MQ MQI client. TLS utiliza este certificado con fines de autenticación. En sistemas AIX, Linux, and Windows , IBM MQ utiliza uno de los valores siguientes para la etiqueta de certificado:

- El valor del atributo de canal o gestor de colas de **CERTLABL** , si está establecido.
- El valor predeterminado de ibmwebspheremq, con el nombre del gestor de colas o el ID de inicio de sesión de usuario de IBM MQ MQI client añadido, todo en minúsculas.

Para obtener más información, consulte Etiquetas de certificado digital.

El nombre del archivo de repositorio de claves consta de una vía de acceso y un nombre de raíz:

- En sistemas AIX and Linux , la vía de acceso predeterminada para un gestor de colas (establecida al crear el gestor de colas) es /var/mqm/qmgrs/*queue_manager_name*/ssl.

  En sistemas Windows , la vía de acceso predeterminada es *MQ_DATA_PATH*\qmgrs*queue_manager_name*\ssl, donde *MQ_DATA_PATH* es la vía de acceso de datos que se selecciona durante la instalación de IBM MQ. Por ejemplo, C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl.

  El nombre de archivo predeterminado es key.kdb. De forma alternativa, puede utilizar su propia vía de acceso y nombre de archivo.

  Si elige su propia vía de acceso o nombre de archivo, establezca los permisos del archivo para controlar estrechamente el acceso al mismo.
- Para un cliente IBM MQ , no hay ninguna vía de acceso o nombre de archivo predeterminado. Controle estrechamente el acceso a este archivo.

No cree repositorios de claves en un sistema de archivos sin soporte para bloqueos a nivel de archivo, por ejemplo, NFS versión 2 en los sistemas Linux.

Para obtener información sobre cómo comprobar y especificar el nombre de archivo de base de datos de claves, consulte “Cambiar la ubicación del repositorio de claves para un gestor de colas en AIX, Linux, and Windows” en la página 306. Puede especificar el nombre del archivo de base de datos de claves antes o después de crear el repositorio de claves.

Puede utilizar los mandatos **runmqakm** (GSKCapiCmd) o V 9.4.0 V 9.4.0 **runmqktool** (keytool) para gestionar repositorios de claves utilizados por IBM MQ. Para obtener más información, consulte "Mandatos runmqakm y runmqktool en AIX, Linux, and Windows" en la página 551.

El ID de usuario que ejecuta los mandatos para gestionar el repositorio de claves debe tener permiso de escritura para el directorio en el que se crea o actualiza el archivo de repositorio de claves. Para un gestor de colas que utiliza el directorio `ssl` predeterminado, el ID de usuario que ejecuta el mandato **runmqakm** o **runmqktool** debe ser miembro del grupo mqm. Para un IBM MQ MQI client, si ejecuta **runmqakm** o **runmqktool** desde un ID de usuario que es diferente al ID de usuario que ejecuta el cliente, debe modificar los permisos de archivo para permitir que el IBM MQ MQI client acceda al repositorio de claves. Para obtener más información, consulte "Acceso y protección de los archivos de base de datos de claves en Windows" en la página 304 o "Acceso y protección de los archivos de base de datos de claves en sistemas AIX and Linux" en la página 304.

Puede crear un repositorio de claves nuevo, vacío, utilizando el mandato **runmqakm** .
V 9.4.0 V 9.4.0 Si utiliza el mandato **runmqktool** en su lugar, el repositorio de claves se crea cuando se emite un mandato para crear o importar un certificado.

Nota: Si tiene que gestionar los certificados TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Procedimiento

1. Emita el mandato siguiente para crear un repositorio de claves con el mandato **runmqakm** :

```
runmqakm -keydb -create -db filename -pw password -type type
         -stash -fips -strong
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-type *tipo*
V 9.4.0 V 9.4.0 Especifica el tipo de repositorio de claves. Para un repositorio de claves utilizado por IBM MQ, los valores posibles son:

- `pkcs12`
- Deprecated `cms`

  Nota: A partir de IBM MQ 9.4.0, el uso de repositorios de claves CMS y archivos de ocultación está en desuso para aplicaciones IBM MQ Java y no está soportado para canales AMQP y MQTT que utilizan SSL/TLS.

-stash
Opcional. Especifique esta opción para almacenar la contraseña del repositorio de claves en un archivo de ocultación. No es necesario almacenar la contraseña en un archivo de ocultación si, en su lugar, cifra la contraseña utilizando el sistema de protección de contraseñas de IBM MQ .

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que están validados con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

-strong
Comprueba que la contraseña especificada cumple los requisitos mínimos de validez de contraseña. Los requisitos mínimos para una contraseña son los siguientes:

- La contraseña debe tener una longitud mínima de 14 caracteres.

- La contraseña debe contener un mínimo de un carácter en minúsculas, un carácter en mayúsculas, y un dígito o un carácter especial. Los caracteres especiales incluyen el asterisco (*), el signo de dólar ($), el signo de número (#) y el signo de porcentaje (%). Un espacio se clasifica como un carácter especial.
- Cada carácter puede aparecer un máximo de tres veces en una contraseña.
- Dos es el número máximo de caracteres consecutivos que pueden ser idénticos.
- Todos los caracteres pertenecen al juego de caracteres ASCII imprimibles estándar dentro del rango entre 0x20 y 0x7e inclusive.

2. Establezca los permisos de acceso para los archivos de repositorio de claves tal como se describe en "Acceso y protección de los archivos de base de datos de claves en Windows" en la página 304 o "Acceso y protección de los archivos de base de datos de claves en sistemas AIX and Linux" en la página 304.

   En Windows, de forma predeterminada solo se otorga acceso al ID de usuario que ha ejecutado el mandato para crear el repositorio de claves para leer el archivo stash (`.sth`). Después de crear un archivo de ocultación con el mandato **runmqakm** , compruebe los permisos de archivo y otorgue permiso a la cuenta de servicio que ejecuta el gestor de colas, o a un grupo como, por ejemplo, `mqm`local.
3. Si no está utilizando un archivo de ocultación, proporcione la contraseña de almacén de claves al gestor de colas o a la aplicación cliente siguiendo las instrucciones de "Suministro de la contraseña del repositorio de claves para un gestor de colas en AIX, Linux, and Windows" en la página 306 o "Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows" en la página 308.

Qué hacer a continuación

Añada certificados de entidad emisora de certificados (CA) predeterminados al repositorio de claves vacío, si es necesario. Para obtener más información, consulte "Adición de certificados de CA predeterminados a un repositorio de claves vacío en AIX, Linux, and Windows" en la página 305.

ALW *Generación de contraseñas fuertes para la protección de repositorios de claves en AIX, Linux, and Windows*

Puede generar contraseñas seguras para la protección de repositorios de claves mediante el mandato **runmqakm** (GSKCapiCmd).

Puede utilizar el mandato **runmqakm** con los siguientes parámetros para generar una contraseña segura:

```
runmqakm -random -create -length password_length -strong -fips
```

donde *longitud_contraseña* es la longitud de la contraseña que se va a generar. La longitud mínima de contraseña que se puede especificar es 14.

Al utilizar la contraseña generada en el parámetro **-pw** de mandatos de administración de certificados posteriores, incluya siempre la contraseña entre comillas dobles. En los sistemas AIX and Linux, también debe utilizar un carácter de barra inclinada invertida para escapar los caracteres siguientes si aparecen en la serie de contraseña:

```
!  \  "  '
```

Cuando especifica una contraseña de repositorio de claves en respuesta a una solicitud del mandato **runmqakm** o V 9.4.0 V 9.4.0 **runmqktool** , no es necesario entrecomillar o escapar la contraseña, ya que el shell del sistema operativo no afecta a la entrada de datos en estos casos.

ALW *Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows*

Varios componentes de IBM MQ necesitan acceso a un repositorio de claves que contenga certificados digitales o claves simétricas. Un repositorio de claves está protegido con una contraseña, ya que contiene información confidencial. La contraseña del repositorio de claves debe almacenarse en una ubicación en

la que IBM MQ pueda leerla cuando se acceda al repositorio de claves. La contraseña también debe estar cifrada para reducir la probabilidad de acceso no autorizado al repositorio de claves.

Los siguientes componentes y características de IBM MQ dan soporte a dos métodos diferentes para almacenar contraseñas de repositorio de claves:

- El repositorio de claves TLS del gestor de colas.
- IBM MQ MQI clients que utilizan TLS.
- V 9.4.0 La configuración de HA nativa en la stanza **NativeHALocalInstance** del archivo `qm.ini` .
- V 9.4.0 La configuración de autenticación de señal en la stanza **AuthToken** del archivo `qm.ini` .

Las contraseñas de repositorio de claves para su uso por parte de estos componentes se pueden cifrar y almacenar utilizando uno de los métodos siguientes:

El sistema de protección por contraseña de IBM MQ .

Cada componente de IBM MQ proporciona un mandato para cifrar la contraseña del repositorio de claves. El mandato cifrado que el mandato genera se almacena en un archivo.

Para el repositorio de claves TLS del gestor de colas, la contraseña se cifra cuando se establece el atributo de gestor de colas **SSLKEYRPWD** .

La contraseña se cifra con el algoritmo AES-128 . Los detalles de este algoritmo son de conocimiento público y se considera seguro.

La contraseña se almacena en un formato propietario que no entiende otro software que pueda acceder al repositorio de claves.

Una contraseña cifrada por un componente IBM MQ no puede ser utilizada por un componente IBM MQ diferente.

Se puede proporcionar una clave de cifrado exclusiva cuando la contraseña del repositorio de claves está cifrada. Una clave de cifrado exclusiva impide que cualquier persona que no tenga acceso a la clave de cifrado pueda descifrar la contraseña.

La contraseña del repositorio de claves de texto sin formato es necesaria para gestionar los certificados que están en el repositorio de claves. Además de cifrar la contraseña del repositorio de claves utilizando el sistema de protección de contraseñas de IBM MQ , también debe almacenar la contraseña del repositorio de claves en una ubicación segura a la que se pueda acceder con este fin.

Para obtener más información sobre el sistema de protección por contraseña de IBM MQ , consulte “Protección de contraseñas en archivos de configuración de componentes de IBM MQ” en la página 575.

Un archivo de ocultación de repositorio de claves.

El mandato **runmqakm** puede almacenar la contraseña del repositorio de claves en un archivo de ocultación.

La contraseña se cifra con un método propietario que es específico del proveedor criptográfico de IBM MQ, IBM Global Security Kit (GSKit).

No se puede proporcionar una clave de cifrado exclusiva.

La contraseña cifrada se almacena en un archivo de ocultación en el mismo directorio que el archivo de repositorio de claves.

Cualquier persona con acceso de lectura al repositorio de claves y al archivo de ocultación puede acceder y gestionar el contenido del repositorio de claves.

Nota: V 9.4.0 Deprecated A partir de IBM MQ 9.4.0, el uso de archivos de ocultación con aplicaciones IBM MQ Java está en desuso.

Importante: V 9.4.0 V 9.4.0 A partir de IBM MQ 9.4.0, los archivos de ocultación no están soportados por los canales AMQP y MQTT que utilizan TLS.

Independientemente del método que elija para cifrar la contraseña del repositorio de claves, asegúrese de que conoce las limitaciones del cifrado de contraseñas almacenadas. Para obtener más información, consulte "Los límites de la protección a través del cifrado de contraseña" en la página 583.

Conceptos relacionados

"Suministro de la contraseña del repositorio de claves para un gestor de colas en AIX, Linux, and Windows" en la página 306
Como el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

"Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows" en la página 308
Como el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

"Trabajar con SSL/TLS en AIX, Linux, and Windows" en la página 299
En sistemas AIX, Linux, and Windows, el soporte TLS (Transport Layer Security) se instala con IBM MQ.

Windows *Acceso y protección de los archivos de base de datos de claves en Windows*

Es posible que los archivos de base de datos de claves no tengan permisos de acceso adecuados. Debe establecer un acceso adecuado a estos archivos.

Establezca el control de acceso a los archivos *key*`.p12`, *key*`.kdb`, *key*`.sth`, *key*`.crl`y *key*`.rdb`, donde *key* es el nombre raíz de la base de datos de claves, para otorgar autorización a un conjunto restringido de usuarios.

Si ha utilizado una extensión de repositorio de claves distinta de `.p12` o `.kdb`, debe asegurarse también de que se hayan establecido los permisos de este archivo.

Considere otorgar acceso del modo siguiente:

autorización total
BUILTIN\Administrators, NT AUTHORITY\SYSTEM y el usuario que creó los archivos de base de datos.

autorización de lectura
Para un gestor de colas, sólo el grupo mqm local. Con esto se presupone que el MCA se está ejecutando con un ID de usuario en el grupo mqm.

Para un cliente, el ID de usuario con el que se está ejecutando el proceso de cliente.

Linux AIX *Acceso y protección de los archivos de base de datos de claves en sistemas AIX and Linux*

Es posible que los archivos de base de datos de claves no tengan permisos de acceso adecuados. Debe establecer un acceso adecuado a estos archivos.

Para un gestor de colas, establezca los permisos en los archivos de bases de datos de claves de manera que el gestor de colas y los procesos de canales puedan leerlos cuando sea necesario pero que otros usuarios no puedan leerlos o modificarlos. Normalmente el usuario mqm necesita permisos de lectura. Si ha creado el archivo de bases de datos de claves iniciando sesión como usuario mqm, es posible que los permisos sean suficientes; si usted no era el usuario mqm sino otro usuario del grupo mqm, tal vez necesite otorgar permisos de lectura a otros usuarios del grupo mqm.

Igual que para un cliente, establezca los permisos en los archivos de las bases de datos de claves de manera que los procesos de la aplicación cliente puedan leerlos cuando sea necesario pero que otros usuarios no puedan leerlos o modificarlos. Normalmente el usuario con el que se ejecuta el proceso de cliente necesita permisos de lectura. Si ha creado el archivo de bases de datos de claves iniciando sesión como dicho usuario, es posible que los permisos sean suficientes; si usted no era el usuario cliente sino otro usuario de dicho grupo, tal vez necesite otorgar permisos de lectura a otros usuarios del grupo.

Establezca los permisos en los archivos *key*.p12, *key*.kdb, *key*.sth, *key*.crly *key*.rdb, donde *key* es el nombre raíz de la base de datos de claves, en read y write para el propietario del archivo, y en read para el grupo de usuarios mqm o cliente (-rw-r -----).

Si ha utilizado una extensión de repositorio de claves distinta de .p12 o .kdb, debe asegurarse también de que se hayan establecido los permisos de este archivo.

ALW *Adición de certificados de CA predeterminados a un repositorio de claves vacío en AIX, Linux, and Windows*

Siga este procedimiento para añadir uno o varios de los certificados de entidad emisora de certificados (CA) predeterminados a un repositorio de claves vacío.

Cuando crea un nuevo repositorio de claves, está vacío. Puede añadir certificados de CA predeterminados a un repositorio de claves utilizando el mandato **runmqakm** .

Utilización de runmqakm

Emita el mandato siguiente para añadir certificados de CA predeterminados a un repositorio de claves con el mandato **runmqakm** :

```
runmqakm -cert -populate -db filename -pw password
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

Nota: IBM MQ confía en todos los certificados firmados por los certificados de CA en el repositorio de claves. Considere detenidamente en qué entidades emisoras de certificados desea confiar y añada sólo los certificados de CA necesarios para autenticar los clientes y los gestores de colas. No se recomienda añadir el conjunto completo de certificados de CA predeterminados a un repositorio de claves.

ALW ***Localizar el repositorio de claves para un gestor de colas en AIX, Linux, and Windows***

Utilice este procedimiento para obtener la ubicación del archivo de base de datos de claves del gestor de colas.

Procedimiento

1. Visualice los atributos del gestor de colas, utilizando cualquiera de los mandatos MQSC siguientes:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

   También puede visualizar los atributos del gestor de colas mediante IBM MQ Explorer o los mandatos PCF.
2. Examine la salida del mandato para localizar la vía de acceso y nombre de raíz del archivo de base de datos de claves.
   Por ejemplo,
   a. En AIX and Linux: /var/mqm/qmgrs/QM1/ssl/key, donde /var/mqm/qmgrs/QM1/ssl es la vía de acceso y key es el nombre de raíz
   b. en Windows: *MQ_INSTALLATION_PATH*\qmgrs\QM1\ssl\key, donde *MQ_INSTALLATION_PATH*\qmgrs\QM1\ssl es la vía de acceso y key es el nombre de raíz. *MQ_INSTALLATION_PATH* representa el directorio de alto nivel en el que está instalado IBM MQ.

Nota: A partir de IBM MQ 9.3.0 , el campo SSLKEYR da soporte a un nombre de archivo completo (incluida la extensión) y a un nombre de raíz (sin extensión). Si se establece un nombre de raíz, IBM MQ añade automáticamente `.kdb` y utiliza ese repositorio de claves.

ALW *Cambiar la ubicación del repositorio de claves para un gestor de colas en AIX, Linux, and Windows*

Puede cambiar la ubicación del archivo de base de datos de claves del gestor de colas de diversas maneras, incluyendo el mandato MQSC ALTER QMGR.

Puede cambiar la ubicación del archivo de base de datos de claves del gestor de colas mediante el mandato MQSC ALTER QMGR para establecer el atributo de repositorio de claves del gestor de colas. Por ejemplo, en AIX and Linux:

```
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/QM1/ssl/MyKey.kdb')
```

En Windows:

```
ALTER QMGR SSLKEYR('C:\Archivos de programa\IBM\MQ\Qmgrs\QM1\ssl\Mykey.kdb')
```

Atención: En Windows y Linux, si se utilizan canales TLS AMQP, el sufijo del archivo de repositorio de claves debe ser uno de los siguientes:

- `.kdb`, para un repositorio de claves CMS
- `.p12` o `.pkcs12`, para un repositorio de claves PKCS #12 .

También puede alterar los atributos del gestor de colas mediante IBM MQ Explorer o los mandatos PCF.

Cuando se cambia la ubicación del archivo de base de datos de claves de un gestor de colas, los certificados no se transfieren desde la ubicación antigua. Si el archivo de base de datos de claves al que está accediendo ahora es un nuevo archivo de base de datos de claves, debe rellenarlo con los certificados de CA y personales que necesita, tal como se describe en “Importación de un certificado personal en un repositorio de claves en AIX, Linux, and Windows” en la página 563.

Suministro de la contraseña del repositorio de claves para un gestor de colas en AIX, Linux, and Windows

Como el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

IBM MQ proporciona dos mecanismos para proporcionar la contraseña del repositorio de claves a un gestor de colas:

- “El atributo KEYRPWD” en la página 306
- “El archivo de ocultación del repositorio de claves” en la página 307

Si no utiliza un archivo de ocultación de repositorio de claves, la contraseña del repositorio de claves se cifra utilizando el sistema de protección de contraseñas de IBM MQ . Para obtener más información sobre los métodos de protección de la contraseña del repositorio de claves, consulte “Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows” en la página 302.

El atributo KEYRPWD

Para proporcionar una contraseña de repositorio de claves directamente al gestor de colas, ejecute el siguiente mandato MQSC, sustituyendo *password* por la contraseña de repositorio de claves:

```
ALTER QMGR KEYRPWD('password')
```

Atención: Asegúrese de rodear la contraseña con comillas simples; de lo contrario, IBM MQ convertirá los caracteres a mayúsculas.

Cuando se especifica una contraseña de repositorio de claves utilizando este método, la contraseña se cifra utilizando el sistema de protección de contraseñas de IBM MQ antes de que se almacene.

Una clave de cifrado, que se conoce como la clave inicial, se utiliza para cifrar la contraseña. Establezca el gestor de colas para que utilice una clave inicial exclusiva para proteger de forma segura la contraseña. Si no proporciona una clave inicial, se utiliza la clave predeterminada.

Asegúrese de que el gestor de colas esté configurado con una clave inicial exclusiva antes de establecer la contraseña del repositorio de claves. Puede modificar la clave inicial utilizando el atributo **INITKEY** en el mandato **ALTER QMGR** . Por ejemplo:

```
ALTER QMGR INITKEY('mykey')
```

Aviso: La modificación de la clave inicial después de establecer la contraseña del repositorio de claves no hace que la contraseña del repositorio de claves se cifre con la nueva clave inicial. Al cambiar la clave inicial sin restablecer también la contraseña del repositorio de claves, IBM MQ no puede descifrar la contraseña del repositorio de claves y, por lo tanto, no puede acceder al repositorio de claves.

Para obtener más información sobre el atributo **KEYRPWD** , consulte KEYRPWD.

El archivo de ocultación del repositorio de claves

Si no se proporciona una contraseña de repositorio de claves al gestor de colas utilizando el atributo **KEYRPWD** , IBM MQ presupone que existe un archivo de ocultación en el mismo directorio que el repositorio de claves. El archivo stash tiene el mismo nombre de raíz que el repositorio de claves, pero tiene la extensión `.sth` .

Se crea un archivo de ocultación de repositorio de claves al mismo tiempo que el repositorio de claves, o posterior, como un mandato **runmqakm** independiente.

Atención: El formato del archivo de ocultación es específico del IBM MQ proveedor criptográfico IBM Global Security Kit (GSKit)y no está disponible en plataformas que utilizan un proveedor criptográfico diferente.

Para crear un archivo de ocultación cuando se crea el repositorio de claves, especifique el parámetro **-stash** . Por ejemplo:

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

donde *passw0rd* es la contraseña del repositorio de claves.

Para crear un archivo de ocultación más adelante, ejecute el mandato siguiente:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

donde *passw0rd* es la contraseña del repositorio de claves.

Conceptos relacionados

“Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows” en la página 302
Varios componentes de IBM MQ necesitan acceso a un repositorio de claves que contenga certificados digitales o claves simétricas. Un repositorio de claves está protegido con una contraseña, ya que contiene información confidencial. La contraseña del repositorio de claves debe almacenarse en una ubicación en la que IBM MQ pueda leerla cuando se acceda al repositorio de claves. La contraseña también debe estar cifrada para reducir la probabilidad de acceso no autorizado al repositorio de claves.

“Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows” en la página 308
Como el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

ALW *Localizar el repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows*

La ubicación del repositorio de claves la proporciona la variable MQSSLKEYR, o se especifica en la llamada MQCONNX.

Examine la variable de entorno MQSSLKEYR para obtener la ubicación del archivo de base de datos de claves del IBM MQ MQI client. Por ejemplo:

```
echo $MQSSLKEYR
```

Compruebe también la aplicación, porque el nombre del archivo de base de datos de claves también se puede establecer en una llamada MQCONNX, según se describe en "Especificación de la ubicación del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows" en la página 308. El valor establecido en una llamada MQCONNX altera temporalmente el valor de MQSSLKEYR.

ALW *Especificación de la ubicación del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows*

No hay ningún repositorio de claves predeterminado para un IBM MQ MQI client. Puede especificar la ubicación del mismo de dos maneras. Asegúrese de que solamente puedan acceder al archivo de base de datos de claves los usuarios o administradores designados para impedir que se realice una copia no autorizada en otros sistemas.

Puede especificar la ubicación del archivo de base de datos de claves para el IBM MQ MQI client de dos formas:

- Estableciendo la variable de entorno MQSSLKEYR. Por ejemplo, en AIX and Linux:

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

  En Windows:

  ```
  set MQSSLKEYR=C:\Archivos de programa\IBM\MQ\ssl\key.kdb
  ```

- Proporcionando la vía de acceso y el nombre de raíz del archivo de base de datos de claves en el campo *KeyRepository* de la estructura MQSCO cuando una aplicación realiza una llamada MQCONNX. Para obtener más información sobre la utilización de la estructura MQSCO en MQCONNX, consulte Visión general de MQSCO.

Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows

Como el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

IBM MQ proporciona cuatro mecanismos para proporcionar la contraseña del repositorio de claves a un IBM MQ MQI client:

- "Los campos KeyRepoPassword de MQSCO " en la página 309
- "La variable de entorno MQKEYRPWD" en la página 309
- "El atributo SSLKeyRepositoryPassword del archivo de configuración del cliente" en la página 309
- "El archivo de ocultación del repositorio de claves" en la página 310

Si no utiliza un archivo de ocultación de repositorio de claves, puede proporcionar la contraseña del repositorio de claves como una serie de texto sin formato, o una serie que se cifre utilizando el sistema de protección de contraseñas de IBM MQ . Para obtener más información sobre los métodos de protección de la contraseña del repositorio de claves, consulte "Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows" en la página 302.

Los campos KeyRepoPassword de MQSCO

Para proporcionar una contraseña de repositorio de claves utilizando la estructura MQSCO, debe utilizar una combinación de los tres campos de serie de variables siguientes:

KeyRepoPasswordLength
La longitud de la contraseña.

KeyRepoPasswordPtr
Puntero a la ubicación en la memoria que contiene la contraseña.

KeyRepoPasswordOffset
La ubicación de la contraseña en la memoria, representada como número de bytes desde el inicio de la estructura MQSCO.

Nota: Sólo puede proporcionar uno de **KeyRepoPasswordPtr** o **KeyRepoPasswordOffset**.

Por ejemplo:

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

Atención: Si proporciona la contraseña utilizando este método, cifre la contraseña antes de que se proporcione a la aplicación IBM MQ client . Para obtener más información, consulte “Cifrado de la contraseña del repositorio de claves” en la página 310.

Para obtener más información sobre la estructura MQCSO, consulte MQSCO-Opciones de configuración SSL/TLS.

La variable de entorno *MQKEYRPWD*

Si no se proporciona una contraseña de repositorio de claves al cliente utilizando la estructura MQSCO, puede especificar la contraseña de repositorio de claves utilizando la variable de entorno *MQKEYRPWD* . Por ejemplo:

```
export MQKEYRPWD=passw0rd
```

o

```
set MQKEYRPWD=passw0rd
```

donde `passw0rd` es la contraseña.

Atención: Si proporciona la contraseña utilizando este método, cifre la contraseña antes de establecer el valor de la variable de entorno. Para obtener más información, consulte “Cifrado de la contraseña del repositorio de claves” en la página 310.

El atributo SSLKeyRepositoryPassword del archivo de configuración del cliente

Si no se proporciona una contraseña del repositorio de claves al cliente utilizando uno de los otros métodos, puede especificar la contraseña del repositorio de claves utilizando el atributo **SSLKeyRepositoryPassword** en la stanza **SSL** del archivo de configuración del cliente. Por ejemplo:

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

Atención: Si proporciona la contraseña utilizando este método, cifre la contraseña antes de establecer el valor del atributo **SSLKeyRepositoryPassword** . Para obtener más información, consulte “Cifrado de la contraseña del repositorio de claves” en la página 310.

Para obtener más información sobre la stanza SSL del archivo de configuración de cliente, consulte Stanza SSL del archivo de configuración de cliente.

El archivo de ocultación del repositorio de claves

Si la contraseña del repositorio de claves no se proporciona al cliente utilizando uno de los otros métodos, IBM MQ presupone que existe un archivo de ocultación en el mismo directorio que el repositorio de claves. El archivo stash tiene el mismo nombre de raíz que el repositorio de claves, pero tiene la extensión `.sth` .

Se crea un archivo de ocultación de repositorio de claves al mismo tiempo que el repositorio de claves, o posterior, utilizando un mandato **runmqakm** independiente.

Atención: El formato del archivo de ocultación es específico del IBM MQ proveedor criptográfico IBM Global Security Kit (GSKit)y no está disponible en plataformas que utilizan un proveedor criptográfico diferente.

Para crear un archivo de ocultación cuando se crea el repositorio de claves, especifique el parámetro **-stash** . Por ejemplo:

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

donde *passw0rd* es la contraseña del repositorio de claves.

Para crear un archivo de ocultación más adelante, ejecute el mandato siguiente:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

donde *passw0rd* es la contraseña del repositorio de claves.

Cifrado de la contraseña del repositorio de claves

Si proporciona la contraseña del repositorio de claves utilizando cualquier método que no sea un archivo de ocultación, cifre la contraseña utilizando el sistema de protección de contraseñas de IBM MQ . Para cifrar la contraseña, ejecute el mandato **runmqicred** . Especifique la contraseña del repositorio de claves cuando se le solicite. El mandato genera la contraseña cifrada. La contraseña cifrada se puede proporcionar a IBM MQ MQI client en lugar de a la contraseña de texto sin formato utilizando cualquiera de los métodos descritos.

Una clave de cifrado, que se conoce como la clave inicial, se utiliza para cifrar la contraseña. Cuando cifre la contraseña, utilice una clave inicial exclusiva para proteger de forma segura la contraseña. Para proporcionar su propia clave inicial, utilice el parámetro **-sf** para el mandato **runmqicred** . Si no proporciona una clave inicial, se utiliza la clave predeterminada.

Para obtener más información, consulte runmqicred (proteger contraseñas de cliente de IBM MQ).

Si proporciona su propia clave inicial cuando la contraseña del repositorio de claves está cifrada y proporciona la contraseña cifrada a IBM MQ MQI client, también debe asegurarse de que proporciona la misma clave inicial a IBM MQ MQI client. Para obtener más información sobre cómo proporcionar la clave inicial a un IBM MQ MQI client, consulte “Suministro de una clave inicial para un IBM MQ MQI client en AIX, Linux, and Windows” en la página 311.

Conceptos relacionados

“Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows” en la página 302
Varios componentes de IBM MQ necesitan acceso a un repositorio de claves que contenga certificados digitales o claves simétricas. Un repositorio de claves está protegido con una contraseña, ya que contiene información confidencial. La contraseña del repositorio de claves debe almacenarse en una ubicación en la que IBM MQ pueda leerla cuando se acceda al repositorio de claves. La contraseña también debe estar cifrada para reducir la probabilidad de acceso no autorizado al repositorio de claves.

“Suministro de la contraseña del repositorio de claves para un gestor de colas en AIX, Linux, and Windows” en la página 306

Como el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

ALW *Suministro de una clave inicial para un IBM MQ MQI client en AIX, Linux, and Windows*
Si proporciona variables a un IBM MQ MQI client que se ha cifrado utilizando el sistema de protección por contraseña de IBM MQ , es posible que tenga que proporcionar la clave inicial correspondiente que se ha utilizado para cifrar el valor.

Si no ha especificado una clave inicial al cifrar el valor, no es necesario que proporcione ningún valor de clave inicial a IBM MQ client. Sin embargo, si ha utilizado una clave inicial exclusiva, puede proporcionar la clave inicial a IBM MQ client utilizando los métodos siguientes:

- "Suministro de la clave inicial utilizando la estructura MQCSP" en la página 311
- "Suministro de la clave inicial utilizando la variable de entorno MQS_MQI_KEYFILE" en la página 311
- "Suministro de la clave inicial utilizando el archivo de configuración de cliente" en la página 312

Suministro de la clave inicial utilizando la estructura MQCSP

Para proporcionar la clave inicial utilizando la estructura MQCSP, debe utilizar una combinación de los tres campos de serie de variables siguientes:

InitialKeyLength
La longitud de la clave inicial

InitialKeyPtr
Un puntero a la ubicación en la memoria que contiene la clave inicial

InitialKeyOffset
La ubicación de la clave inicial en la memoria, representada como número de bytes desde el inicio de la estructura MQCSP.

Nota: Sólo puede proporcionar uno de **InitialKeyPtr** o **InitialKeyOffset**.

Por ejemplo:

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

Suministro de la clave inicial utilizando la variable de entorno MQS_MQI_KEYFILE

Si no se proporciona una clave inicial al cliente utilizando la estructura MQCSP, IBM MQ comprueba la variable de entorno *MQS_MQI_KEYFILE* . Debe establecer esta variable de entorno en la ubicación de un archivo que contenga una sola línea de texto, que conste de la clave inicial que desea utilizar.

Por ejemplo, si existe un archivo denominado `mykey.key` en el directorio raíz y contiene la clave inicial, debe establecer la variable de entorno como se indica a continuación:

```
export MQS_MQI_KEYFILE=/mykey.key
```

o

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

Suministro de la clave inicial utilizando el archivo de configuración de cliente

Si no se proporciona una clave inicial al cliente utilizando un mecanismo anterior, IBM MQ comprueba el atributo **MQIInitialKeyFile** de la stanza Security del archivo `mqclient.ini` . Debe establecer este atributo en la ubicación de un archivo que contenga una sola línea de texto, que conste de la clave inicial que desea utilizar.

Por ejemplo, si existe un archivo denominado `mykey.key` en el directorio raíz y contiene la clave inicial, el archivo de configuración del cliente debe contener lo siguiente:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

Conceptos relacionados

"Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows" en la página 308
Como el repositorio de claves contiene información confidencial, está protegido con una contraseña. Para poder acceder al contenido del repositorio de claves para realizar operaciones TLS, IBM MQ debe poder recuperar la contraseña del repositorio de claves.

"Trabajar con SSL/TLS" en la página 281
Estos temas proporcionan instrucciones para realizar tareas individuales relacionados con la utilización de TLS con IBM MQ.

ALW *Cuándo entran en vigor los cambios en los certificados o en el repositorio de claves en AIX, Linux, and Windows*

Cuando cambia los certificados en un repositorio de claves, o la ubicación del repositorio de claves, los cambios entran en vigor en un momento que depende del tipo de canal y de cómo se ejecuta el canal.

Los cambios en los certificados en el repositorio de claves, o en la ubicación del repositorio de claves, entran en vigor en las situaciones siguientes:

- Cuando un nuevo proceso de canal de salida individual ejecuta por primera vez un canal TLS.
- Cuando un nuevo proceso de canal de entrada individual TCP/IP recibe por primera vez una solicitud para iniciar un canal TLS.
- Cuando se emite el mandato MQSC **REFRESH SECURITY TYPE(SSL)** para renovar el entorno TLS.
- Para los procesos de la aplicación cliente, cuando se cierra la última conexión TLS del proceso. La siguiente conexión TLS recuperará los cambios del certificado.
- Para canales que se ejecutan como hebras de un proceso de agrupación de procesos (amqrmppa), cuando se inicia o se reinicia el proceso de agrupación de procesos y ejecuta por primera vez un canal TLS. Si el proceso de agrupación de procesos ya ha ejecutado un canal TLS y desea que el cambio entre en vigor inmediatamente, ejecute el mandato MQSC **REFRESH SECURITY TYPE(SSL)**.
- Para canales que se ejecutan como hebras del iniciador de canal, cuando se inicia o se reinicia el iniciador de canal y ejecuta por primera vez un canal TLS. Si el proceso iniciador de canal ya ha ejecutado un canal TLS y desea que el cambio entre en vigor inmediatamente, ejecute el mandato MQSC **REFRESH SECURITY TYPE(SSL)**.
- Para canales que se ejecutan como hebras de un escucha TCP/IP, cuando se inicia o se reinicia el escucha y recibe por primera vez una solicitud para iniciar un canal TLS. Si el escucha ya ha ejecutado un canal TLS y desea que el cambio entre en vigor inmediatamente, ejecute el mandato MQSC **REFRESH SECURITY TYPE(SSL)**.

También puede renovar el entorno TLS de IBM MQ utilizando los mandatos IBM MQ Explorer o PCF.

Importante: Los cambios en el archivo de configuración del almacén de claves, o en el almacén de claves utilizado por un interceptor de MCA de Advanced Message Security (AMS) o un cliente de AMS , entran en vigor cuando se reinicia el gestor de colas o la aplicación.

ALW *Configuración del hardware de cifrado en AIX, Linux, and Windows*

Puede configurar el hardware de cifrado para un gestor de colas o cliente de varias maneras.

Puede configurar hardware de cifrado para un gestor de colas en AIX, Linux, and Windows mediante uno de los dos métodos siguientes:

- Utilice el mandato MQSC de **ALTER QMGR** con el parámetro SSLCRYP , tal como se describe en ALTER QMGR.
- Utilice IBM MQ Explorer para configurar el hardware de cifrado en el sistema AIX, Linux, and Windows . Para obtener más información, consulte la ayuda en línea.

Puede configurar el hardware criptográfico para un cliente IBM MQ en AIX, Linux, and Windows utilizando uno de los métodos siguientes:

- Establezca la variable de entorno **MQSSLCRYP** . Los valores permitidos para **MQSSLCRYP** son los mismos que para el parámetro **SSLCRYP** , tal como se describe en ALTER QMGR. Para establecer esta variable de entorno, utilice uno de estos mandatos:
  - Linux AIX En sistemas AIX and Linux:

    ```
    export MQSSLCRYP=string
    ```

  - Windows En sistemas Windows:

    ```
    SET MQSSLCRYP=string
    ```

  donde *string* representa la serie de parámetro que se utilizará para configurar el hardware de cifrado presente en el sistema.

  Si utiliza la versión GSK_PKCS11 del parámetro **SSLCRYP** , la etiqueta de señal PKCS #11 debe coincidir con la etiqueta con la que ha configurado el hardware.
- Establezca el atributo **SSLCryptoHardware** en la stanza SSL del archivo de configuración IBM MQ client . Los valores permitidos son los mismos que para el parámetro **SSLCRYP** , tal como se describe en **ALTER QMGR**.

  Si utiliza la versión GSK_PKCS11 del parámetro **SSLCRYP** , la etiqueta de señal PKCS #11 debe coincidir con la etiqueta con la que ha configurado el hardware.
- Establezca el campo **CryptoHardware** de la estructura de opciones de configuración SSL, MQSCO, en una llamada MQCONNX. Si desea más información, consulte Visión general de MQSCO.

Atención: >Al proporcionar la configuración para el hardware de cifrado a través de la variable de entorno **MQSSLCRYP** , o el atributo **SSLCryptoHardware** , debe proteger la contraseña antes de almacenarla. Para obtener más información, consulte “IBM MQ clients que utilizan hardware criptográfico” en la página 579.

Si ha configurado hardware de cifrado que utiliza la interfaz PKCS #11 utilizando cualquiera de estos métodos, debe almacenar el certificado personal para utilizarlo en sus canales en el archivo de base de datos de claves del señal de cifrado que ha configurado. Este tema se describe en el apartado “Gestión de certificados en el hardware PKCS #11” en la página 573.

MQ Appliance Trabajar con SSL/TLS en IBM MQ Appliance

IBM MQ Appliance tiene soporte de TLS (Transport Layer Security).

IBM MQ Appliance tiene diferentes mandatos para gestionar certificados. Para obtener información detallada sobre la gestión de certificados, consulte la documentación de IBM MQ Appliance, Gestión de certificados 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 314.

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 de ID de usuario adicionales para TLS en z/OS*

Esta información describe los requisitos adicionales que su ID de usuario necesita para configurarse y funcionar con TLS en z/OS.

Asegúrese de que tiene todas las actualizaciones generales y de alto impacto (HIPER) en el sistema.

Si el repositorio de claves es propiedad del ID de usuario CHINIT, este ID de usuario necesita acceso de lectura al IRR IRR.DIGTCERT.LISTRING en la clase FACILITY y, de lo contrario, actualice el acceso y el acceso de lectura a la IRR de IRR.DIGTCERT.LIST . Otorgue acceso utilizando el mandato PERMIT con ACCESS (UPDATE) o ACCESS (READ) según corresponda.

Asegúrese de que ha configurado los siguientes prerrequisitos:

- El ID de usuario *ssidCHIN* se ha definido correctamente en RACFy el ID de usuario *ssidCHIN* tiene el acceso adecuado a los perfiles siguientes.
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

  Estas variables están definidas en la clase FACILITY de RACF.
- El ID de usuario *ssidCHIN* es el propietario del conjunto de claves.
- El certificado personal del gestor de colas, si lo ha creado el mandato RACDCERT, se crea con un ID de usuario de tipo de certificado igual al del ID de usuario *ssidCHIN*.
- El iniciador de canal se recicla, o se emite el mandato **REFRESH SECURITY TYPE(SSL)** , para recoger los cambios que realice en el conjunto de claves.
- El procedimiento iniciador de canal de IBM MQ tiene acceso a la biblioteca de ejecución de SSL del sistema *pdsname*.SIEALNKE a través de la lista de enlaces, LPA o de una sentencia STEPLIB DD. Esta biblioteca debe estar autorizada para APF.
- El ID de usuario bajo cuya autorización se ejecuta el iniciador de canal está configurado para utilizar z/OS UNIX System Services (z/OS UNIX), tal como se describe en la documentación de z/OS UNIX System Services Planning .

  Los usuarios que no desean que el iniciador de canal invoque z/OS UNIX utilizando el UID y el segmento OMVS, sólo necesitan modelar un nuevo segmento OMVS basado en el segmento predeterminado, ya que el iniciador de canal no requiere permisos especiales y no se ejecuta en UNIX como superusuario.

  Consulte los mandatos PERMIT en "Giving the channel initiator the correct access rights on z/OS" en la página 316 para obtener algunos ejemplos sobre cómo otorgar al iniciador de canal el acceso correcto.

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 “Repositorio de claves 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 digitales” 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 268

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 315.
- *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 317. 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 "Etiquetas de certificados digitales, descripción de los 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 319.

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 315.
- *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 315.
- *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 319. 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 autenticación 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 319.
- 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 321.

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 y CipherSuites en 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 323, 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.

Identificación y autenticación de usuarios

Puede identificar y autenticar usuarios utilizando certificados X.509 , la estructura MQCSP o en varios tipos de programa de salida de usuario.

Utilización de certificados X.509

Puede identificar y autenticar usuarios utilizando certificados X.509 con el mandato **SET CHLAUTH** y el parámetro **SSLPEER** . El parámetro **SSLPEER** especifica un filtro a utilizar para compararlo con el Nombre distinguido del sujeto del certificado del cliente o gestor de colas igual en el otro extremo del canal.

Para obtener más información sobre cómo utilizar el mandato **SET CHLAUTH** y el parámetro **SSLPEER** , consulte SET CHLAUTH.

Las Entidades emisoras de certificados pueden revocar los certificados digitales. Puede comprobar el estado de revocación de los certificados utilizando OCSP, o listas de revocación de certificados (CRL) en servidores LDAP, dependiendo de la plataforma. Para obtener más información, consulte “Trabajar con certificados revocados” en la página 346.

Utilización de la estructura MQCSP

La estructura de parámetros de seguridad de conexión MQCSP se especifica en una llamada MQCONNX. Esta estructura puede contener credenciales proporcionadas por la aplicación. La aplicación puede proporcionar un ID de usuario y una contraseña en la estructura MQCSP. A partir de IBM MQ 9.3.4, las aplicaciones también pueden proporcionar una señal de autenticación. Si es necesario, MQCSP se puede modificar en una salida de seguridad.

Aviso: Las credenciales de una estructura MQCSP a veces se envían a través de la red en texto sin formato. Para asegurarse de que las credenciales de aplicación cliente están protegidas, consulte “Protección por contraseña MQCSP” en la página 32.

Para obtener más información, consulte “Identificación y autenticación de usuarios utilizando la estructura MQCSP” en la página 328 y “Cómo trabajar con señales de autenticación” en la página 331.

Linux AIX En AIX y Linux, el ID de usuario y la contraseña especificados en la estructura MQCSP se pueden autenticar utilizando el sistema operativo o el Método de autenticación conectable (PAM). PAM proporciona un mecanismo general para la autenticación de usuarios que oculta los detalles de los servicios. Para obtener más información, consulte “Utilización de PAM (Pluggable Authentication Method)” en la página 358.

Implementación de identificación y autenticación en salidas

Puede identificar y autenticar usuarios utilizando varios tipos de programa de salida de usuario. Para obtener más información, consulte “Implementación de la identificación y autenticación en salidas de seguridad” en la página 329, “Correlación de identidad en salidas de mensajes” en la página 330 y “Correlación de identidad en la salida de API y la salida cruzada de API” en la página 330.

Usuarios privilegiados

Un usuario privilegiado es aquel que tiene autorización administrativa completa para IBM MQ.

Además de los usuarios listados en la tabla siguiente, hay ciertos objetos y autorizaciones a los que hay que prestar una atención especial a la hora de otorgar acceso, para garantizar la integridad y la seguridad del gestor de colas. Cuando se otorgue cualquiera de las autorizaciones siguientes, hay que prestar una atención especial:

- Cualquier autorización a un objeto SYSTEM
- Autorizaciones de administración para crear, alterar y suprimir objetos.

  z/OS En z/OS, esta es la autorización de seguridad de mandatos y la autorización de seguridad de recursos de mandato para emitir mandatos DEFINE, ALTER y DELETE.

  Multi En todas las demás plataformas, estas autorizaciones son autorizaciones de administración como, por ejemplo, `+crt`, `+chg` y `+dlt`.
- Autorización de administración para borrar colas.

z/OS En z/OS, esta es la autorización de seguridad de mandatos y la autoridad de seguridad de recursos de mandato para emitir mandatos CLEAR.

Multi En todas las demás plataformas, esta autorización es `+clr`.

- Las autorizaciones de administración para detener canales, restituir o confirmar mensajes.

  z/OS En z/OS, esta autorización es de seguridad de mandatos y la autorización de seguridad de recursos de mandatos para emitir mandatos como, por ejemplo, RESET CHANNEL, START CHANNEL y STOP CHANNEL.

  Multi En todas las demás plataformas, estas autorizaciones son `+ctrl` y `+ctrlx`.
- La autorización MQI de usuario alternativo que permite a las aplicaciones escalar privilegios para comprobaciones de autorización.

  z/OS En z/OS, esta autorización es cualquier autorización otorgada a los perfiles de seguridad de usuario alternativo.

  Multi En todas las demás plataformas, esta autorización es `+altusr`.
- Las autorizaciones de contexto que permiten a las aplicaciones cambiar el contexto de seguridad de los mensajes.

  z/OS En z/OS, esta autorización es cualquier autorización otorgada a los perfiles de seguridad de contexto.

  Multi En todas las demás plataformas, estas autorizaciones son `+setall` y `+setid`.

Como regla general, a las aplicaciones de mensajería solo se les debería otorgar las autorizaciones MQI en las colas o temas que sean necesarios. Los canales MCA que ejecutan con un MCAUSER sin privilegios y algunos otros tipos de aplicaciones especiales como, por ejemplo, manejadores de colas de mensajes no entregados, pueden requerir autorizaciones adicionales que no suelen otorgarse a las aplicaciones para que funcionen correctamente.

Tabla 67. Usuarios privilegiados por plataforma

Plataforma	**Usuarios privilegiados**
Sistemas Windows	• SISTEMA • Miembros del grupo mqm • Miembros del grupo Administradores
Sistemas AIX and Linux	• Miembros del grupo mqm
Sistemas IBM i	• Los perfiles qmqm y qmqmadm • Todos los miembros del grupo qmqmadm • Cualquier usuario definido con el valor *ALLOBJ
z/OS	El ID de usuario bajo el que se ejecutan el iniciador de canal, el gestor de colas y los espacios de direcciones de seguridad de mensajes avanzados. Estos ID de usuario no tienen automáticamente las autorizaciones administrativas completas para IBM MQ, pero se consideran privilegiados debido al nivel de acceso que normalmente suelen recibir estos ID de usuario.

Identificación y autenticación de usuarios utilizando la estructura MQCSP

Puede especificar la estructura de parámetros de seguridad de conexión MQCSP en una llamada MQCONNX. La estructura MQCSP es el método principal para que las aplicaciones que utilizan la interfaz de cola de mensajes (MQI) controlen las credenciales que se utilizan para la autenticación.

La estructura MQCSP contiene credenciales que el servicio de autorización puede utilizar para identificar y autenticar el usuario.

Las salidas de seguridad del lado del cliente o del servidor pueden modificar la estructura MQCSP, incluso si la aplicación no proporciona explícitamente la estructura MQCSP. Un ejemplo de aplicación que no proporciona explícitamente una estructura MQCSP es una aplicación que utiliza IBM MQ classes for JMS. Para ver un ejemplo de una salida de seguridad del lado del cliente que inserta un ID de usuario y una contraseña en la estructura MQCSP, consulte "Salida de seguridad del lado del cliente para insertar ID de usuario y contraseña (mqccred)" en la página 84.

V 9.4.0 La estructura MQCSP contiene un ID de usuario y una contraseña, o una señal de autenticación. Las restricciones siguientes se aplican a las credenciales proporcionadas en la estructura MQCSP:

- Una aplicación o salida debe proporcionar un ID de usuario y una contraseña, o una señal de autenticación, pero no ambos.
- Sólo se pueden utilizar señales de autenticación que cumplan determinados formatos y requisitos para acceder a IBM MQ. Para obtener más información sobre los requisitos para las señales de autenticación en IBM MQ, consulte "Requisitos para las señales de autenticación" en la página 334.
- Si la identidad en la señal de autenticación se va a adoptar como contexto para la aplicación, la señal debe proporcionar una reclamación de usuario adecuada y el valor de reclamación debe ser un ID de usuario IBM MQ válido. Por ejemplo, el nombre de usuario debe cumplir con las restricciones de longitud máxima y caracteres especiales. Para obtener más información sobre la adopción de un ID de usuario, consulte "Relación entre los valores MQCSP y ADOPTCTX" en la página 328.

Para obtener más información sobre la estructura MQCSP, consulte MQCSP-Parámetros de seguridad.

Aviso: Las credenciales de una estructura MQCSP para una aplicación cliente se envían a veces a través de la red en texto sin formato. Para asegurarse de que las credenciales de aplicación cliente están protegidas, consulte "Protección por contraseña MQCSP" en la página 32.

Relación entre los valores MQCSP y ADOPTCTX

IBM MQ siempre autentica las credenciales que se pasan en la estructura MQCSP si la característica de autenticación de conexión está habilitada. Después de que las credenciales se hayan autenticado correctamente, IBM MQ puede adoptar el ID de usuario para comprobaciones de autorización posteriores en las operaciones realizadas por la aplicación conectada. El ID de usuario en las credenciales MQCSP se adopta si el objeto de información de autenticación (AUTHINFO) al que hace referencia el atributo **CONNAUTH** del gestor de colas se define con **ADOPTCTX(YES)**.

IBM MQ tiene un límite en la longitud de los ID de usuario que puede utilizar para comprobaciones de autorización. Para obtener más información sobre estos límites, consulte "ID de usuario" en la página 93. Cuando se adopta un ID de usuario pasado en la estructura MQCSP, IBM MQ se comporta de forma diferente, en función de otras opciones de configuración:

- Cuando se utiliza la autenticación de conexión LDAP, IBM MQ adopta el ID de usuario que está en el atributo de nombre de usuario corto del registro LDAP del usuario. El atributo de nombre de usuario corto se establece utilizando el atributo **SHORTUSR** del objeto AUTHINFO.

  Por ejemplo, si **SHORTUSR** se establece en `'CN'`, y el registro LDAP lista el usuario como `'CN=Test,SN=MQ,O=IBM,C=UK'`, se utiliza el ID de usuario `Test`.
- Cuando se utiliza la autenticación de conexión de sistema operativo o la autenticación PAM, si ADOPTCTX es YES, el ID de usuario pasado en la estructura MQCSP se trunca para cumplir el límite de ID de usuario de 12 caracteres de IBM MQ cuando se adopta como contexto de conexión.

Si **ChlAuthEarlyAdopt** está habilitado, el truncamiento se produce después de que se hayan autenticado las credenciales de usuario.

Si **ChlAuthEarlyAdopt** no está habilitado, el truncamiento se produce antes de la adopción. En Windows, si el usuario se proporciona con el formato `user@domain`, esto significa que el truncamiento puede dar como resultado una especificación de dominio que no es válida cuando el usuario tiene menos de 12 caracteres.

Por ejemplo, si se proporciona un usuario `` `ibmmq@windowsdomain` `` a través de MQCSP, se trunca en `` `ibmmq@window` `` en este escenario. Esto da como resultado el siguiente error:

```
AMQ8074W: La autorización ha fallado porque el SID 'SID' no coincide con la entidad
'ibmmq@window'
```

Sobre esta base, si pasa un ID de usuario de más de 12 caracteres, como un ID de usuario de dominio de Windows con el formato `user@domain`, a través de MQCSP debe configurar **ChlAuthEarlyAdopt**=*Y* en el archivo qm.ini para evitar este error.

De forma alternativa, utilice ADOPTCTX (NO) en la configuración CONNAUTH AUTHINFO y utilice un enfoque alternativo como, por ejemplo, una regla CHLAUTH USERMAP, una salida de seguridad o el valor MCAUSER del objeto de canal para establecer el ID de usuario para el canal.

Implementación de la identificación y autenticación en salidas de seguridad

Puede utilizar una salida de seguridad para implementar autenticación unidireccional o mutua

El principal objetivo de una salida de seguridad es permitir que el MCA de cada extremo de un canal autentique su asociado. En cada extremo de un canal de mensajes, y en el extremo del servidor de un canal MQI, un MCA suele actuar en nombre del gestor de colas al que está conectado. En el extremo del cliente de un canal MQI, un MCA suele actuar en nombre del usuario de la aplicación IBM MQ MQI client. En esta situación, la autenticación mutua realmente tiene lugar entre dos gestores de colas, o entre un gestor de colas y el usuario de una aplicación IBM MQ MQI client.

La salida de seguridad proporcionada (la salida de canal SSPI) ilustra cómo se puede implementar la autenticación mutua intercambiando señales de autenticación que genera, y posteriormente comprueba, un servidor de autenticación fiable como Kerberos. Para obtener más detalles, consulte "El programa de salida de canal SSPI en Windows" en la página 162.

La autenticación mutua también se puede implementar utilizando la tecnología de Infraestructura de claves públicas (PKI). Cada salida de seguridad genera algunos datos aleatorios, los firma utilizando la clave privada del gestor de colas o del usuario al que representa y envía los datos firmados a su asociado en un mensaje de seguridad. La salida de seguridad del asociado lleva a cabo la autenticación comprobando la firma digital mediante la clave pública del gestor de colas o usuario. Antes de intercambiar firmas digitales, es posible que las salidas de seguridad tengan que acordar el algoritmo para generar un resumen de mensaje, en el caso de que se pueda utilizar más de un algoritmo.

Cuando la salida de seguridad envía datos firmados a su asociado, también tiene que enviar algún medio de identificar el gestor de colas o usuario al que representa. Puede ser un Nombre distinguido o incluso un certificado digital. Si se envía un certificado digital, la salida de seguridad del asociado puede validar el certificado trabajando a través de una cadena de certificados hasta el certificado de CA raíz. Esto asegura la propiedad de la clave pública que se utiliza para comprobar la firma digital.

La salida de seguridad del asociado sólo puede validar un certificado digital si tiene acceso a un repositorio de claves que contiene los demás certificados de la cadena de certificados. Si no se envía un certificado digital correspondiente al gestor de colas o al usuario, debe haber uno disponible en el repositorio de claves al que la salida de seguridad del asociado tenga acceso. La salida de seguridad del asociado no puede comprobar la firma digital a no ser que encuentre la clave pública del firmante.

TLS (seguridad de la capa de transporte) utiliza técnicas PKI como las que se acaban de describir. Para obtener más información sobre cómo SSL lleva a cabo la autenticación, consulte "Conceptos de TLS (Transport Layer Security)" en la página 19.

Si no está disponible ningún servidor de autenticación fiable ni el soporte para PKI, se pueden utilizar otras técnicas. Una técnica común, que se puede implementar en salidas de seguridad, utiliza un algoritmo de clave simétrica.

Una de las salidas de seguridad, la salida A, genera un número aleatorio y lo envía en un mensaje de seguridad a su salida de seguridad asociada, la salida B. La salida B cifra el número utilizando su copia de una clave que sólo conocen las dos salidas de seguridad. La salida B envía el número cifrado a la salida A en un mensaje de seguridad con un segundo número aleatorio que ha generado la salida B. La salida A verifica que el primer número aleatorio se ha cifrado correctamente, cifra el segundo número aleatorio utilizando su copia de la clave y envía el número cifrado a la salida B en un mensaje de seguridad. Luego la salida B verifica que el segundo número aleatorio se ha cifrado correctamente. Durante este intercambio, si alguna de las salidas de seguridad no está satisfecha con la autenticidad de la otra, puede indicar al MCA que cierre el canal.

Una ventaja de esta técnica es que no se envía ninguna clave ni contraseña a través de la conexión de comunicaciones durante el intercambio. Una desventaja es que no proporciona una solución al problema de cómo distribuir la clave compartida de forma segura. Una solución a este problema se describe en "Implementación de confidencialidad en programas de salida de usuario" en la página 476. Una técnica parecida se utiliza en SNA para la autenticación mutua de dos LU cuando se vinculan para formar una sesión. La técnica se describe en "Autenticación a nivel de sesión" en la página 128.

Todas las técnicas anteriores para la autenticación mutua se pueden adaptar para proporcionar autenticación unidireccional.

Correlación de identidad en salidas de mensajes

Puede utilizar salidas de mensajes para procesar información para autenticar un ID de usuario, aunque puede ser mejor implementar la autenticación a nivel de aplicación.

Cuando una aplicación transfiere un mensaje a una cola, el campo *UserIdentifier* del descriptor de mensaje contiene un ID de usuario asociado a la aplicación. Sin embargo, no hay datos que se puedan utilizar para autenticar el ID de usuario. Estos datos se pueden añadir mediante una salida de mensajes en el extremo emisor de un canal y se pueden comprobar mediante una salida de mensajes en el extremo receptor del canal. Los datos de autenticación pueden ser una contraseña cifrada o una firma digital, por ejemplo.

Este servicio puede resultar más eficaz si se implementa a nivel de aplicación. El requisito básico es que el usuario de la aplicación que recibe el mensaje pueda identificar y autenticar al usuario de la aplicación que ha enviado el mensaje. Por lo tanto es natural considerar la implementación de este servicio a nivel de aplicación. Para obtener más información, consulte "Correlación de identidad en la salida de API y la salida cruzada de API" en la página 330.

Correlación de identidad en la salida de API y la salida cruzada de API

Una aplicación que recibe un mensaje debe poder identificar y autenticar al usuario de la aplicación que ha enviado el mensaje. Este servicio normalmente se implementa mejor a nivel de aplicación. Las salidas de API pueden implementar el servicio de varias maneras.

En cuanto a un mensaje individual se refiere, la identificación y autenticación son un servicio en el que participan dos usuarios, el emisor y el receptor del mensaje. El requisito básico es que el usuario de la aplicación que recibe el mensaje pueda identificar y autenticar al usuario de la aplicación que ha enviado el mensaje. Tenga en cuenta que el requisito de autenticación es unidireccional y no bidireccional.

Dependiendo de cómo se implemente, es posible que los usuarios y sus aplicaciones necesiten una interfaz o deban interactuar con el servicio. Además, cuándo y cómo se utiliza el servicio dependerá de dónde están ubicados los usuarios y sus aplicaciones y de la naturaleza de las aplicaciones propiamente dichas. Por lo tanto, es natural considerar la implementación del servicio a nivel de aplicación en lugar de a nivel de enlace.

Si piensa implementar este servicio a nivel de enlace, es posible que deba tener en cuenta algunos aspectos como, por ejemplo, los siguientes:

- Cómo aplicará el servicio, en un canal de mensajes, solamente a los mensajes que lo requieren
- Cómo permitirá que los usuarios y las aplicaciones se comuniquen o interactúen con el servicio, si éste es un requisito
- Dónde invocará los componentes del servicio en una situación de varios saltos, en la que se envía un mensaje a través de más de un canal hasta llegar a su destino

A continuación se muestran algunos ejemplos de cómo se puede implementar el servicio de identificación y autorización a nivel de aplicación. El término *salida de API* significa una salida de API o una salida cruzada de API.

- Cuando una aplicación transfiere un mensaje a una cola, una salida de API puede obtener una señal de autenticación de un servidor de autenticación fiable, como Kerberos. La salida de API puede añadir esta señal a los datos de aplicación del mensaje. Cuando la aplicación receptora recupera el mensaje, una segunda salida de API puede solicitar al servidor de autenticación que autentique al emisor comprobando la señal.
- Cuando una aplicación transfiere un mensaje a una cola, se puede añadir una salida de API a los elementos siguientes de los datos de aplicación del mensaje:
  - El certificado digital del emisor
  - La firma digital del emisor

  Si se dispone de algoritmos diferentes para generar un resumen del mensaje, la salida de API puede incluir el nombre del algoritmo que ha utilizado.

  Cuando la aplicación receptora recupera el mensaje, una segunda salida de API puede realizar las comprobaciones siguientes:

  - La salida de API puede validar el certificado digital analizando la cadena de certificados hasta llegar al certificado de la CA raíz. Para hacerlo, la salida de API debe tener acceso al repositorio de claves que contiene los certificados restantes de la cadena de certificados. Esta comprobación asegura que el emisor, identificado mediante el Nombre distinguido, es el propietario genuino de la clave pública que contiene el certificado.
  - La salida de API puede comprobar la firma digital utilizando la clave pública que contiene el certificado. Esta comprobación autentica al emisor.

  El Nombre distinguido del emisor se puede enviar en lugar del certificado digital completo. En este caso, el repositorio de claves debe contener el certificado del emisor, de modo que la segunda salida de API pueda buscar la clave pública del emisor. Otra posibilidad es enviar todos los certificados de la cadena de certificados.
- Cuando una aplicación transfiere un mensaje a una cola, el campo *UserIdentifier* del descriptor de mensaje contiene un ID de usuario asociado a la aplicación. El ID de usuario se puede utilizar para identificar al emisor. Para habilitar la autenticación, una salida de API puede añadir algunos datos como, por ejemplo, una contraseña cifrada, a los datos de la aplicación que contiene el mensaje. Cuando la aplicación receptora recupera el mensaje, una segunda rutina de API puede autenticar el ID de usuario utilizando los datos que ha transportado el mensaje.

  Esta técnica puede considerarse suficiente en los mensajes que se originan en un entorno controlado y fiable, y en aquellas circunstancias en las que no se disponga de un servidor de autenticación fiable o de soporte para PKI.

Linux V 9.4.0 AIX Cómo trabajar con señales de autenticación

A partir de IBM MQ 9.4.0, las aplicaciones cliente pueden proporcionar señales para autenticarse con un gestor de colas que se ejecuta en AIX o Linux. El ID de usuario de la señal también se puede utilizar para la autorización para acceder a los recursos de IBM MQ .

Las JWT (JSON Web Tokens) adoptan un modelo de identidad basado en reclamaciones. La identidad y el control de acceso se abstraen en ideas de reclamaciones y emisores de señales.

- Una reclamación es un par nombre-valor que contiene información sobre un usuario y establece quién es el usuario, no lo que puede hacer.

- El emisor de señal es un tercero de confianza o un servidor que emite una señal para un usuario basándose sólo en la identidad del usuario. El emisor de señal no está preocupado por lo que puede hacer el usuario.

Una señal es una estructura simple que contiene reclamaciones y se puede transferir fácilmente entre partes a través de Internet. El uso de señales para la autenticación tiene la ventaja de la gestión de identidades centralizada. Puede utilizar un emisor de señal de confianza para que las aplicaciones puedan autenticarse con muchos servicios sin registrarse por separado con cada servicio. Las señales proporcionan una mayor seguridad ya que las credenciales no se envían a cada servicio, sólo al emisor de confianza.

Un JWT se define a través del estándar de Internet propuesto RFC7519.

Cómo funcionan las señales con IBM MQ

Las señales que se utilizan con IBM MQ deben ser JWT válidas que se hayan firmado con un algoritmo al que IBM MQ dé soporte. El JWT debe estar firmado de acuerdo con el estándar de firma web JSON (JWS). Las señales que utilizan las tecnologías JWE (JSON Web Encryption) y JWK (JSON Web Key) JOSE no se pueden utilizar con IBM MQ. Para obtener más información, consulte “Requisitos para las señales de autenticación” en la página 334.

La aplicación que proporciona la señal de autenticación se puede ejecutar en cualquier plataforma que dé soporte a IBM MQ clients. La aplicación debe estar escrita en C o en Java, y conectarse al gestor de colas utilizando enlaces de cliente. Sin embargo, el gestor de colas debe ejecutarse en AIX o Linux.

El gestor de colas valida la firma de señal contra la clave pública o la clave simétrica del emisor de confianza en el repositorio de claves. Para configurar el gestor de colas, siga los pasos de “Configuración de un gestor de colas para aceptar señales de autenticación utilizando un punto final JWKS” en la página 337 o Configuración de un gestor de colas para aceptar señales de autenticación utilizando un almacén de claves local.

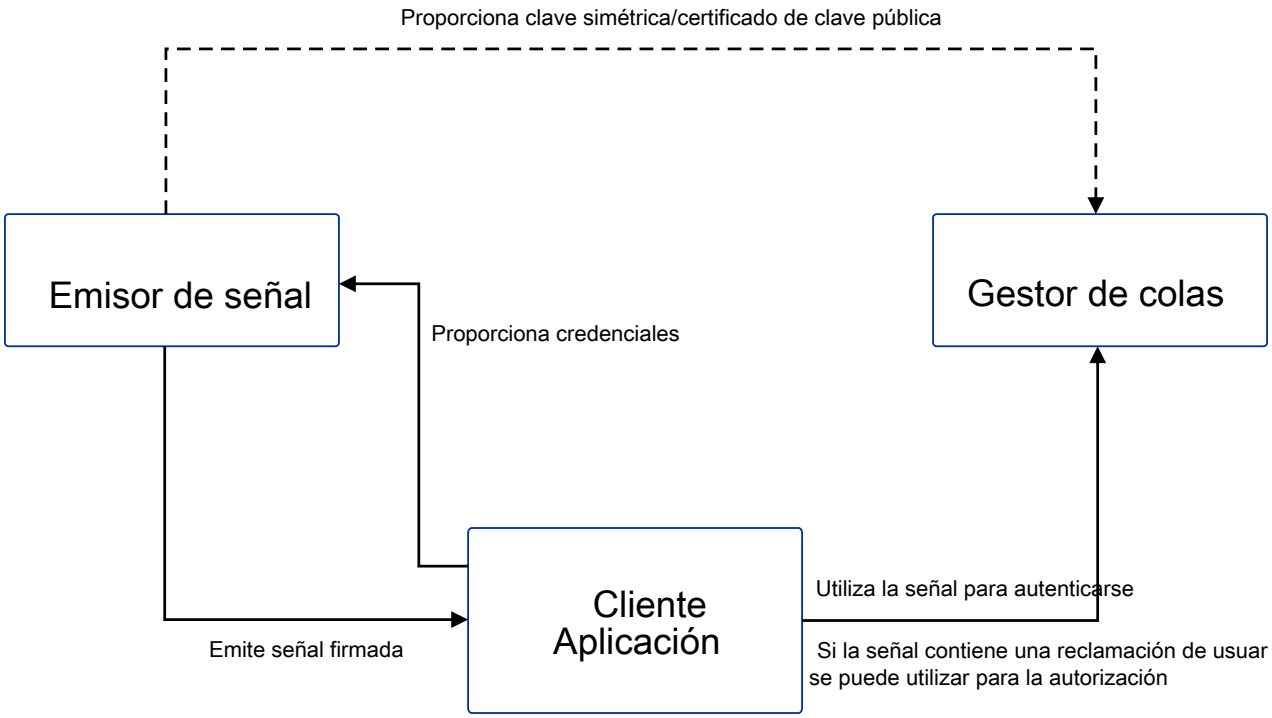
El emisor de señal es la parte de confianza que tiene el acceso de seguridad delegado, lo que significa que verifican la identidad del usuario de la aplicación. El gestor de colas comprueba que una señal de autenticación es válida y que el usuario autenticado tiene autorización para acceder a objetos IBM MQ . El gestor de colas puede, pero no necesita saber de los usuarios antes de que se conecten por primera vez con una señal. El administrador de IBM MQ debe configurar la autenticación y la autorización para las aplicaciones que se conectan al gestor de colas y establecer los requisitos para lo que deben contener las señales.

La aplicación cliente puede solicitar dinámicamente una señal del emisor que utiliza para la autenticación cuando se conecta a IBM MQ. A continuación, la aplicación utiliza la estructura MQCSP, o el equivalente en la API elegida, para pasar la señal al gestor de colas cuando se conecta.

Si la aplicación no se puede cambiar para solicitar una señal de autenticación y presentar la señal al gestor de colas cuando se conecta, también se puede utilizar una salida de seguridad para proporcionar una señal en la estructura MQCSP.

Si la señal cumple los requisitos para las señales de autenticación y la firma de señal es válida, se establece la conexión. El gestor de colas también puede utilizar el ID de usuario contenido en la señal para comprobaciones de autorización para acceder a los recursos de IBM MQ si la reclamación de usuario opcional está contenida en la señal. La reclamación de usuario es la reclamación dentro de la señal que contiene el ID de usuario que el gestor de colas adopta para las comprobaciones de autorización. Este nombre de la reclamación de usuario se especifica con el atributo **`UserClaim`** en la stanza **`AuthToken`** del archivo `qm.ini` .

Para obtener más información, consulte “Utilización de señales de autenticación en una aplicación” en la página 342 y MQCSP-Parámetros de seguridad.



El diagrama muestra un ejemplo básico del flujo esperado para el uso de señales con IBM MQ. El ciclo de vida esperado es el siguiente:

- El emisor de confianza emite la señal a una aplicación. Para obtener más información, consulte Requisitos para señales de autenticación.
- La aplicación pasa la señal al gestor de colas al conectarse. Para obtener más información, consulte Utilización de señales de autenticación en una aplicación.
- El gestor de colas valida la firma de señal contra la clave pública o la clave simétrica del emisor de confianza en el repositorio de claves. Para configurar el gestor de colas, siga los pasos de “Configuración de un gestor de colas para aceptar señales de autenticación utilizando un punto final JWKS” en la página 337.
- Si la señal de autenticación contiene una reclamación de usuario válida, el usuario de la señal se puede adoptar para comprobaciones de autorización para acceder a los recursos de IBM MQ . Para obtener más información, consulte Adopción de usuarios para autorización.
- El administrador de IBM MQ gestiona los certificados de emisor de señales de confianza. Cuando el certificado caduca, se debe obtener un nuevo certificado del emisor de señales y añadirlo al repositorio de claves.
- Si ha configurado el gestor de colas y la aplicación se está conectando pero encuentra problemas con la señal, consulte Resolución de problemas de la señal de autenticación y Códigos de error de autenticación de señal.

IBM MQ funciona con cualquier emisor de señales que proporcione señales que se ajusten a los estándares JWT y JWS.

Si todavía no está utilizando señales pero desea comprender qué implica poner en marcha un servidor de señales, consulte la Guía de iniciación para el proyecto Keycloak gratuito y de código abierto.

Referencia relacionada

Stanza AuthToken del archivo `qm.ini`

Linux V 9.4.0 AIX Requisitos para las señales de autenticación

Requisitos de validación, estructura y algoritmos para las señales de autenticación utilizadas con IBM MQ.

Requisitos

Las señales de autenticación que se utilizan con IBM MQ deben cumplir los requisitos siguientes.

- La longitud del símbolo no debe superar la longitud máxima de 8192 caracteres. Para obtener más información, consulte TokenLength (MQLONG) para MQCSP.
- La estructura y codificación de señal es válida tal como la define la especificación JWT (JSON Web Token) en RFC7519y la especificación JWS (JSON Web Signature) en RFC7515.
- Los parámetros de cabecera de señal necesarios que se especifican en Tabla 68 en la página 335 están presentes y los valores de los parámetros son válidos.
- Las reclamaciones de carga útil necesarias especificadas en Tabla 69 en la página 336 están presentes y los valores de las reclamaciones son válidos.
- La señal se firma con un algoritmo en Tabla 70 en la página 336 al que IBM MQ da soporte.
- El valor de la reclamación de caducidad (**exp**) es posterior a la hora actual.
- Si la reclamación no antes de (**nbf**) está presente, el valor es anterior a la hora actual.
- Si existe una reclamación de usuario, el valor debe cumplir los requisitos de “ID de usuario en señales de autenticación” en la página 337.

Estructura de señal

IBM MQ acepta JWT que se ajustan al estándar RFC7519 . El JWT debe estar firmado y codificado de acuerdo con el estándar JWS definido en RFC7515.

IBM MQ espera que la señal protegida JWS contenga los tres componentes siguientes:

Cabecera JOSE

Un objeto JSON que contiene parámetros que describen el tipo de señal y los algoritmos criptográficos que se utilizan para proteger su contenido.

El ejemplo de cabecera siguiente declara que el objeto codificado es un JWT y que la cabecera y la carga útil están protegidas utilizando el algoritmo HMAC SHA-256 .

```
{
 "typ":"JWT",
 "alg":"HS256"
}
```

Carga útil de JWS

Un objeto JSON que contiene reclamaciones tal como se especifica en el estándar JWT. Cada miembro del objeto JSON es una reclamación. Las reclamaciones pueden confirmar la identidad del emisor de señal o el ID de usuario del portador.

```
{
 "exp": 1685529153,
 "nbf": 1685528150,
 "AppUser": "MyUserName"
 }
```

Firma JWS

Se utiliza para validar que la señal la emite un emisor de confianza.

Estos componentes se representan en la señal protegida JWS como series base64url-encoded separadas por un punto ('.').

Una señal de autenticación que se ajusta al estándar JWS se firma para permitir que se valide la autenticidad de la señal, pero no se cifra. Por lo tanto, puede ser leído, y posiblemente reutilizado, por cualquiera que tenga acceso a la señal. Configure la conexión con el gestor de colas para asegurarse de que la autenticación está protegida utilizando el cifrado cuando se envía a través de la red, por ejemplo, utilizando TLS. Para obtener más información sobre las opciones para proteger las credenciales proporcionadas por una aplicación, consulte Protección de contraseña MQCSP.

IBM MQ da soporte a los siguientes parámetros y reclamaciones en la cabecera y la carga útil de las señales de autenticación. Los parámetros o reclamaciones adicionales de una señal se ignoran. Si una señal contiene más de un parámetro o reclamación con el mismo nombre, se utiliza el último parámetro o reclamación con el nombre duplicado.

Tabla 68. Descripciones de parámetros de cabecera de señal

Parte de señal	Nombre de parámetro	Tipo de datos	Obligatorio	Descripción
Cabecera	**typ**	Serie	Sí	Tipo de señal. El valor de este parámetro debe ser "JWT".
	alg	Serie	Sí	El algoritmo utilizado para proteger la cabecera y la carga útil. El valor de este parámetro debe ser uno de los algoritmos de Tabla 70 en la página 336.

Tabla 69. Descripciones de reclamaciones de carga útil de señal				
Parte de señal	**Nombre de parámetro**	**Tipo de datos**	**Obligatorio**	**Descripción**
Carga útil	**exp**	Entero	Sí	La hora de caducidad de la señal, expresada como el número de segundos desde el 1 de enero de 1979, 00:00 Hora Universal Coordinada. La señal no se acepta después de este tiempo.
	nbf	Entero	No	La hora, expresada como el número de segundos desde el 1 de enero de 1979, 00:00 Hora Universal Coordinada antes de la cual no se acepta la señal.
	El nombre de reclamación de usuario ha especificado el campo **UserClaim** de la stanza **AuthToken** en el archivo `qm.ini` .	Serie	Sólo es necesario si la reclamación de usuario en la señal se utiliza para la autorización.	El nombre de la reclamación que contiene el ID de usuario que se adopta para las comprobaciones de autorización. Por ejemplo, si la señal tiene la reclamación de usuario `"AppUser": "MyUserName"`, debe especificar **UserClaim**=`AppUser` en la stanza **AuthToken** del archivo `qm.ini` .

Para ver un buen ejemplo de una señal codificada y descodificada, consulte la página depurador en el sitio web de `jwt.io` .

Algoritmos

IBM MQ da soporte a un subconjunto de algoritmos que se incluyen en la especificación JWA (JSON Web Algorithm) para señales protegidas JWS .

Tabla 70. Algoritmos web JSON (JWA) soportados por IBM MQ para señales protegidas JWS	
alg valor de parámetro	**Firma digital o algoritmo MAC**
HS256	HMAC con SHA-256
HS384	HMAC con SHA-384
HS512	HMAC con SHA-512
RS256	RSASSA-PKCS1-v1_5 utilizando SHA-256
RS384	RSASSA-PKCS1-v1_5 utilizando SHA-384
RS512	RSASSA-PKCS1-v1_5 utilizando SHA-512

Requisitos de certificado de clave asimétrica

Si una señal se firma con una clave asimétrica, el certificado de clave pública del emisor de señales debe estar en el repositorio de claves que el gestor de colas utiliza para la autenticación de señales. Cuando se recibe la señal de autenticación, el certificado debe estar dentro de su periodo de validez. No se realizan comprobaciones para asegurarse de que el certificado del emisor de señal no se ha revocado.

ID de usuario en señales de autenticación

Si el gestor de colas está configurado para adoptar el ID de usuario que está contenido en la reclamación de usuario de una señal de autenticación como contexto para la aplicación, el ID de usuario que se adopta debe cumplir los requisitos siguientes:

- Puede contener hasta 12 caracteres.
- Debe empezar con uno de los siguientes caracteres:

  A-Z a-z
- Puede contener cualquiera de los caracteres siguientes:

  0-9 A-Z a-z +,-. : = _
- No debe ser uno de los ID de usuario reservados UNKNOWN y NOBODY.

Tareas relacionadas

Configuración de un gestor de colas para aceptar **AuthTokens**

Referencia relacionada

Stanza AuthToken del archivo `qm.ini`

Linux V 9.4.0 AIX Configuración de un gestor de colas para aceptar señales de autenticación utilizando un punto final JWKS

Configure el gestor de colas de IBM MQ que se ejecuta en AIX o Linux para autenticar usuarios y aplicaciones con señales de autenticación utilizando un punto final JWKS.

Antes de empezar

Para obtener más información sobre cómo funcionan las señales con IBM MQ, consulte Trabajar con señales de autenticación.

Antes de configurar el gestor de colas, compruebe que el objeto AUTHINFO al que se hace referencia en el atributo **CONNAUTH** del gestor de colas es del tipo `IDPWOS`. La autenticación de señal sólo está disponible cuando el gestor de colas está configurado para la comprobación de ID de usuario y contraseña del sistema operativo.

Compruebe que el atributo **SecurityPolicy** de la stanza Service no esté establecido en `Group`. La autenticación de señal no está disponible si **SecurityPolicy** se establece explícitamente en `Grupo`. Si**SecurityPolicy** se establece en `Grupo` , eliminar el**SecurityPolicy** atributo de la sección Servicio y, a continuación, reinicie el gestor de colas.

Acerca de esta tarea

Las aplicaciones pueden autenticarse con el gestor de colas utilizando señales. IBM MQ acepta señales web JSON (*JWT*) de emisores de confianza que siguen el estándar de Internet propuesto RFC7519. Puede utilizar señales para autenticar una identidad, que luego se puede adoptar para futuras comprobaciones de autorización.

La forma más sencilla de configurar el gestor de colas para que acepte señales es apuntar a un punto final JWKS tal como se describe a continuación. Si el servicio de autenticación no lo proporciona y el punto final o JWKS no es adecuado por otras razones, consulte “Configuración de un gestor de colas para aceptar señales de autenticación utilizando un almacén de claves local” en la página 338.

Procedimiento

1. Solicite al administrador del servidor de autenticación estos detalles:
   - El punto final JWKS correcto (URL).
   - Qué certificado utiliza este servidor para cifrar el tráfico HTTP y/o qué autoridad firma este certificado.

Importante: Siempre debe proporcionar información de JWKS a través de TLS/HTTPS y necesita esta información para asegurarse de que el gestor de colas puede confiar en la conexión.

2. Configure el gestor de colas para crear conexiones https salientes proporcionando un **HTTPSKeyStore** en el archivo qm.ini .

   Para obtener más información, consulte

   - La explicación HTTPSKeyStore en el archivo `qm.ini` .
   - "Creación de un repositorio de claves para utilizarlo como almacén de confianza TLS" en la página 345.

   Si el servidor de autenticación utiliza un certificado/CA a medida, debe asegurarse de que esté correctamente presente en este HTTPSKeyStore.

3. Configure el punto final JWKS definiendo una stanza JWKS en el archivo de configuración qm.ini .

   La stanza adicional proporciona lo siguiente:

   - **`issuername`**. Debe coincidir con la reclamación '`iss`' que está presente en las señales firmadas por esta autoridad, y a menudo se basa en el URL del servicio de autenticación.
   - **`endpoint`**. Es la dirección desde la que el gestor de colas consulta las claves públicas utilizadas para validar firmas de señal.
   - **`userclaim`**.Esto es opcional para identificar un campo personalizado en señales que se debe utilizar para las comprobaciones de autorización de IBM MQ una vez que se ha validado una señal.

     Atención: Debe estar presente si tiene previsto utilizar **ADOPTCTX**(YES) para dichas conexiones.

4. Una vez que se hayan completado los cambios del archivo `.ini` , emita el mandato REFRESH `SECURITY TYPE(AUTHINFO)` o reinicie el gestor de colas.

   Si la configuración es satisfactoria, las aplicaciones pueden conectarse utilizando señales firmadas inmediatamente.

   Si hay algún problema, por ejemplo, no puede ponerse en contacto con el servicio de autenticación para recuperar claves públicas, los problemas se notifican en el archivo de registro AMQERR01 para el gestor de colas.

Resultados

Ha configurado correctamente un gestor de colas para aceptar señales de autenticación utilizando un punto final JWKS.

Nota: Las claves se renuevan periódicamente desde el servidor de autenticación (cada 15 minutos), y con más frecuencia si una aplicación que se conecta presenta un ID de clave desconocido. Normalmente, esto significa que no se necesitan más acciones de configuración de IBM MQ para actualizar los certificados a medida que caducan y se sustituyen en el lado del servidor. Para forzar una renovación inmediata, emita el mandato `REFRESH SECURITY TYPE(AUTHINFO)` en cualquier momento.

Conceptos relacionados

Resolución de problemas de la señal de autenticación

Tareas relacionadas

Utilización de señales de autenticación en una aplicación

Referencia relacionada

Stanza AuthToken del archivo `qm.ini`

Linux V 9.4.0 AIX

Configuración de un gestor de colas para aceptar señales de autenticación utilizando un almacén de claves local

Configure el gestor de colas de IBM MQ para autenticar usuarios y aplicaciones con señales de autenticación.

Antes de empezar

Siempre que sea posible, considere la posibilidad de utilizar un punto final JWKS, consulte "Configuración de un gestor de colas para aceptar señales de autenticación utilizando un punto final JWKS" en la página 337, en lugar de configurar manualmente los certificados de validación de señal. El uso de JWKS normalmente simplifica la configuración inicial y el mantenimiento continuado.

Lea sobre cómo funcionan las señales con IBM MQ en Trabajar con señales de autenticación.

Antes de configurar el gestor de colas, compruebe que el objeto AUTHINFO al que se hace referencia en el atributo **CONNAUTH** del gestor de colas es del tipo IDPWOS. La autenticación de señal sólo está disponible cuando el gestor de colas está configurado para la comprobación de ID de usuario y contraseña del sistema operativo.

Compruebe que el atributo **SecurityPolicy** de la stanza Service no esté establecido en Group. La autenticación de señal no está disponible si **SecurityPolicy** se establece explícitamente en Grupo. Si **SecurityPolicy** se establece en Grupo, elimine el atributo **SecurityPolicy** de la stanza Service y, a continuación, reinicie el gestor de colas.

Acerca de esta tarea

Desde IBM MQ 9.3.4 , las aplicaciones pueden autenticarse con el gestor de colas utilizando señales. IBM MQ acepta señales web JSON (*JWT*) de emisores de confianza que siguen el estándar de Internet propuesto RFC7519. Puede utilizar señales para autenticar una identidad, que luego se puede adoptar para futuras comprobaciones de autorización.

Configure el gestor de colas para que acepte señales guardando el certificado de clave pública o la clave simétrica del emisor de confianza en el depósito de claves del gestor de colas. Añada la stanza AuthToken al archivo qm.ini y renueve la configuración de seguridad para que el gestor de colas recoja la nueva configuración.

Es posible que desee configurar un almacén de claves local en lugar de utilizar JWKS en un entorno de prueba, o cuando no sea posible la conectividad directa con el servidor de autenticación desde el gestor de colas. También puede definir un almacén de claves local además de cualquier punto final JWKS.

Nota: Cuando tanto un punto final JWKS como un almacén de claves local proporcionan un emisor y un KID coincidentes para una señal presentada, se utiliza preferentemente la clave proporcionada por el punto final JWKS.

En estas situaciones, configure el almacén de claves local como se indica a continuación:

Procedimiento

1. Cree el repositorio de claves.
   a) Cree un repositorio de claves para el certificado de clave pública o clave simétrica que se recibe del emisor de confianza. Puede utilizar un repositorio de claves CMS con la extensión de archivo .kdb o un repositorio de claves PKCS#12 con la extensión de archivo .p12.

      Emita el mandato siguiente para crear un repositorio de claves CMS :

      ```
      runmqakm -keydb -create -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
      -type cms
      ```

      Si el**runmqakm** El comando devuelve un error, ver ejecutarmqakm -keydb . Si el mandato se completa correctamente, utilice el mandato ls para listar el contenido del directorio:

      ```
      ls -l /var/mqm/qmgrs/qm1/tokenissuer
      ```

      Se muestran los archivos siguientes:

      ```
      -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) Si es necesario, cambie la propiedad del grupo para los archivos de repositorio de claves que ha creado para que se pueda otorgar acceso de lectura al grupo mqm. Inicialmente, sólo el usuario administrativo que ha ejecutado el mandato tiene acceso a los archivos creados.

```
chgrp mqm /var/mqm/qmgrs/qm1/tokenissuer/key.*
```

c) Cambie la modalidad de los archivos de repositorio de claves para añadir permisos de lectura para el grupo mqm. Por ejemplo, el mandato siguiente añade permisos de lectura/escritura para el propietario del archivo y permisos de sólo lectura para el grupo.

```
chmod 640 /var/mqm/qmgrs/qm1/tokenissuer/key.*
```

2. Cifre la contraseña del repositorio de claves con el mandato **runqmcred** y guarde la serie cifrada en un archivo.

a) Cree un archivo que contenga la clave inicial que se utiliza para cifrar la contraseña del repositorio de claves.

El archivo debe contener la clave inicial como una sola línea de texto. La longitud máxima de la clave inicial es de 256 bytes. Si ya ha establecido una clave inicial para el gestor de colas utilizando el atributo de gestor de colas **INITKEY** , copie el valor del atributo **INITKEY** en el nuevo archivo. Si todavía no ha establecido una clave inicial para el gestor de colas, cree una nueva clave de cifrado exclusiva y añádala al archivo de claves inicial.

Nota: Para obtener más información, consulte INITKEY. Si no especifica la clave inicial, se utiliza una predeterminada. Es más seguro utilizar su propia clave inicial.

Nota: Otorgue los permisos mínimos necesarios en el archivo de claves inicial para mantener el contenido del archivo seguro. El archivo de claves inicial sólo se utiliza para cifrar la contraseña del repositorio de claves. Por lo tanto, sólo los administradores que utilizan la clave inicial para cifrar contraseñas necesitan acceder al archivo de claves inicial de lectura.

b) Si la clave inicial del gestor de colas todavía no está establecida, establezca el valor del atributo **INITKEY** del gestor de colas en la clave inicial que ha creado en el paso “2.a” en la página 340. Utilice el mandato **ALTER QMGR** para establecer la clave inicial del gestor de colas. Por ejemplo:

```
ALTER QMGR INITKEY('myEncrypt10nK3y')
```

c) Emita el mandato **runqmcred** para cifrar la contraseña del repositorio de claves. Utilice el parámetro **-sf** para especificar la vía de acceso al archivo que contiene la clave inicial.

```
runqmcred -sf initial.key
```

Cuando se le solicite, especifique la contraseña del repositorio de claves. El mandato genera la contraseña cifrada.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Enter password:
*************
<QM>!2!b5rb01sMzFzc1ClZeQMryruWFM3HSm8DKyEaZK7qzWY=!TrWdU57DCDXM0Qah99I/Lg==
```

Copie la serie en la última línea y guárdela en un archivo.

3. Utilice uno de los métodos siguientes para añadir el certificado de clave pública o clave simétrica del emisor de señales al repositorio de claves.

- Para añadir el certificado de clave pública RSA al repositorio de claves, emita el mandato siguiente:

```
runmqakm -cert -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile
```

- Para añadir una clave simétrica codificada en base64 al repositorio de claves, emita el mandato siguiente:

```
runmqakm -secretkey -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
-label keylabel
         -file keyfile -format ascii
```

Donde *keylabel* es la etiqueta que se va a adjuntar al certificado o clave secreta, y *keyfile* es el nombre del archivo que contiene el certificado o la clave secreta codificada en base64 .

4. Añada la stanza **AuthToken** y los atributos siguientes al archivo `qm.ini` :
   - La vía de acceso al repositorio de claves, especificada utilizando el atributo **KeyStore** .
   - El archivo que contiene la contraseña para el repositorio de claves, especificado utilizando el atributo **KeyStorePwdFile** .
   - La etiqueta del certificado o clave simétrica que ha añadido en el paso “3” en la página 340, especificado utilizando el atributo **CertLabel** .

   Por ejemplo:

   ```
   AuthToken:
      KeyStore=/var/mqm/qmgrs/qm1/tokenissuer/key.kdb
      KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
      CertLabel=rsakey
   ```

   Donde `key.kdb` es el nombre del repositorio de claves que ha creado en el paso “1.a” en la página 339, y `key.pw` es el archivo que contiene la contraseña cifrada para el repositorio de claves que ha creado en el paso “2.c” en la página 340.

   Para obtener más información sobre la stanza **AuthToken** , consulte la sección AuthToken del archivo `qm.ini`.
5. Si el gestor de colas está configurado para adoptar el ID de usuario contenido en la reclamación de usuario de señal para su uso en comprobaciones de autorización posteriores, añada el atributo **UserClaim** a la stanza **AuthToken** .

   Para determinar si el gestor de colas está configurado para adoptar el ID de usuario en la señal, emita el siguiente mandato MQSC:

   ```
   DISPLAY AUTHINFO(authinfo_name) ADOPTCTX
   ```

   Donde *authinfo_name* es el valor del atributo **CONNAUTH** del gestor de colas. Si el valor del atributo **ADOPTCTX** es YES, el gestor de colas se configura para adoptar el ID de usuario en la señal y el atributo **UserClaim** se debe especificar en la stanza **AuthToken** .

   Establezca el valor del atributo **UserClaim** en el nombre de la reclamación de señal que contiene el ID de usuario que se va a adoptar. Por ejemplo, si la señal contiene la reclamación `"AppUser": "MyUserName"`, añada la línea siguiente a la stanza **AuthToken** :

   ```
   UserClaim=AppUser
   ```
6. Renueve la configuración de seguridad del gestor de colas para que recoja la configuración de señal del archivo `qm.ini` . Emita el mandato siguiente para iniciar el mandato **runmqsc** :

   ```
   runmqsc qm1
   ```

   A continuación, emita el siguiente mandato MQSC:

   ```
   REFRESH SECURITY TYPE(CONNAUTH)
   ```

Qué hacer a continuación

Trabaje con los desarrolladores para ayudarles a comprender cómo pueden utilizar señales en aplicaciones para autenticarse con el gestor de colas.

Conceptos relacionados

Resolución de problemas de la señal de autenticación

Tareas relacionadas

Utilización de señales de autenticación en una aplicación

Referencia relacionada

Stanza AuthToken del archivo `qm.ini`

Linux V 9.4.0 AIX Obtención de una señal de autenticación del emisor de señales elegido

Escriba la aplicación para obtener una señal de autenticación del emisor de señales elegido cuando se conecta a un gestor de colas de IBM MQ .

Antes de empezar

Consulte la información de "Utilización de señales de autenticación en una aplicación" en la página 342.

Procedimiento

- El modo en que obtiene una señal de autenticación y el contenido exacto de la señal varía entre distintos emisores de señales.

  Escriba la aplicación para interactuar con el emisor de señales elegido para solicitar y obtener la señal de autenticación. La señal de autenticación debe cumplir los requisitos de IBM MQ para las señales de autenticación. Para obtener más información sobre estos requisitos, consulte "Requisitos para las señales de autenticación" en la página 334.

  Si tiene previsto adoptar un ID de usuario contenido en una reclamación de señal como contexto para la aplicación, la señal de autenticación también debe cumplir los requisitos siguientes:

  – La señal de autenticación debe contener una reclamación que coincida con el nombre de reclamación de usuario en la configuración de autenticación de señales del gestor de colas.

  – El valor de la reclamación de usuario debe cumplir los requisitos para los ID de usuario en las señales de autenticación. Para obtener más información, consulte "ID de usuario en señales de autenticación" en la página 337.

Resultados

Ahora ha obtenido un JWT con formato correcto que se puede presentar a IBM MQ para su validación.

Tareas relacionadas

Configuración de un gestor de colas para aceptar **AuthTokens**

Referencia relacionada

Stanza AuthToken del archivo `qm.ini`

MQCSP-Parámetros de seguridad

Linux V 9.4.0 AIX Utilización de señales de autenticación en una aplicación

Escriba la aplicación para proporcionar una señal de autenticación cuando se conecte a un gestor de colas de IBM MQ .

Antes de empezar

A partir de IBM MQ 9.4.0, las aplicaciones pueden proporcionar una señal de autenticación cuando se conectan a un gestor de colas.

La solicitud debe cumplir los siguientes requisitos:

- Debe escribirse en C o Java (utilizando IBM MQ classes for JMS/ Jakarta Messaging)
- Debe conectarse al gestor de colas como un IBM MQ client. Es decir, la aplicación debe conectarse al gestor de colas a través de una red, en lugar de utilizar enlaces locales.

- Debe conectarse a un gestor de colas que se ejecute en AIX o Linux.

Si la aplicación no cumple estos requisitos, la conexión falla y el código de razón MQRC_FUNCTION_NOT_SUPPORTED (2298) se devuelve a la aplicación.

La aplicación que proporciona la señal de autenticación se puede ejecutar en cualquier plataforma que dé soporte a IBM MQ MQI clients.

Los clientes que utilizan la reconexión automática de cliente no pueden proporcionar una señal de autenticación cuando se conectan. Si una aplicación proporciona una señal de autenticación y especifica la opción MQCNO_RECONNECT o MQCNO_RECONNECT_Q_MGR en la estructura MQCNO, la conexión falla y el código de razón MQRC_RECONNECT_INCOMPATIBLE (2547) se devuelve a la aplicación. Para obtener más información sobre la reconexión automática de cliente, consulte Reconexión automática de cliente.

Si no puede escribir la aplicación para proporcionar una señal de autenticación debido a estos requisitos, también puede migrar la aplicación para utilizar señales de autenticación utilizando una salida de seguridad de cliente. La salida de seguridad de cliente se puede escribir para establecer la señal de autenticación en la estructura MQCSP. Para obtener más información sobre las salidas de seguridad, consulte Exits de seguridad en una conexión de cliente.

A partir de IBM MQ 9.4.0, las aplicaciones cliente de JMS pueden proporcionar directamente una señal al conectarse (consulte “Obtención de una señal de autenticación del emisor de señales elegido” en la página 342). Antes de IBM MQ 9.4.0, las aplicaciones Java pueden proporcionar indirectamente una señal mediante un programa de salida. Para obtener más información, consulte Clase Java MQCSP.

Acerca de esta tarea

Nota: Una señal de autenticación que se ajusta al estándar JWS (JSON Web Signature) está firmada para permitir que se valide la autenticidad de la señal, pero no está cifrada. Por lo tanto, puede ser leído, y posiblemente reutilizado, por cualquiera que tenga acceso a la señal. Configure la conexión con el gestor de colas para asegurarse de que la señal de autenticación está protegida utilizando el cifrado cuando se envía a través de la red, por ejemplo, utilizando TLS. Para obtener más información sobre las opciones para proteger las credenciales proporcionadas por una aplicación, consulte “Protección por contraseña MQCSP” en la página 32.

Antes de modificar las aplicaciones para conectarse utilizando una señal, asegúrese de que:

- El gestor de colas se ha configurado para aceptar señales de autenticación siguiendo los pasos de “Configuración de un gestor de colas para aceptar señales de autenticación utilizando un almacén de claves local” en la página 338
- La aplicación puede obtener una señal válida según sea necesario del servidor de autenticación, consulte “Obtención de una señal de autenticación del emisor de señales elegido” en la página 342.

Para proporcionar una señal de autenticación cuando la aplicación se conecta a un gestor de colas IBM MQ , incluya el proceso siguiente.

Procedimiento

- Para proporcionar una señal de autenticación desde una aplicación C (MQI):

  La aplicación debe conectarse utilizando MQCONNX (en lugar de MQCONN) y proporcionar una estructura MQCSP :

  – El campo **AuthenticationType** debe establecerse en MQCSP_AUTH_ID_TOKEN.
  – La versión de la estructura debe establecerse en MQCSP_VERSION_3.
  – El campo **TokenPtr** o **TokenOffset** debe hacer referencia a la señal de autenticación.
  – El campo **TokenLength** debe establecerse en la longitud de la señal de autenticación.

  Ejemplo de código C para conectarse a un gestor de colas utilizando MQCSP Versión 3 y señal de autenticación:

```
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 proporcionar una señal de autenticación desde una aplicación Java :

  Las aplicaciones que utilizan IBM MQ classes for JMS/Jakarta Messaging pueden proporcionar una señal a través de cualquiera de los métodos `createContexto createConnection` , que toman un nombre de usuario y una contraseña.

  Para proporcionar una señal de autenticación, el:

  – **UserID** debe establecerse en nulo o en una serie vacía, es decir, sin espacios, " "

  – La señal se proporciona como la serie **Password** .

  Esto se aplica a todas las implementaciones de IBM MQ de la interfaz `ConnectionFactory` .

  Se pueden utilizar los formatos de parámetro explícitos, por ejemplo, `createContext`(String **userID**, String **password**), o las versiones de parámetro implícitas, por ejemplo, `createContext`().

  En el último caso, primero se deben haber proporcionado **userID** y Token **Password** vacíos como propiedades en la fábrica de conexiones.

  Ejemplo de código Java para conectarse a un gestor de colas utilizando una señal de autenticación:

```
// 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
```

  Si la conexión falla con el código de razón MQRC_NOT_AUTHORIZED (2035) o MQRC_SECURITY_ERROR (2063), busque en el registro de errores del gestor de colas un mensaje de error que contenga más información sobre la causa de la anomalía. Para obtener más ayuda con el diagnóstico de problemas con señales de autenticación, consulte Resolución de problemas de señales de autenticación.

Resultados

La aplicación está ahora conectada al gestor de colas. Permanece conectado hasta que se desconecta, incluso si la señal que se ha utilizado para autenticarse caduca. Si la aplicación se desconecta del gestor de colas y necesita volver a conectarse, es posible que tenga que obtener una nueva señal de autenticación con un tiempo de caducidad posterior antes de que se pueda volver a conectar.

Tareas relacionadas

Configuración de un gestor de colas para aceptar **AuthTokens**

Referencia relacionada

Stanza AuthToken del archivo `qm.ini`

MQCSP-Parámetros de seguridad

Linux V 9.4.0 AIX Creación de un repositorio de claves para utilizarlo como almacén de confianza TLS

Al crear conexiones TLS salientes, debe crear un 'almacén de confianza ' simple que pueda validar certificados firmados por un conjunto común de entidades emisoras de certificados (CA). Las conexiones TLS de ejemplo son un canal de cliente IBM MQ o una conexión HTTPS, tal como se utiliza al configurar algunos componentes de IBM MQ.

Acerca de esta tarea

Atención: Decidir qué certificados y entidades emisoras de certificados deben confiar en el entorno es un paso importante con implicaciones para la seguridad de la configuración de extremo a extremo. Este tema se proporciona para ilustrar los pasos comunes que permiten a los componentes de IBM MQ confiar en el mismo conjunto de certificados ya configurados para el sistema operativo; sin embargo, si tiene dudas, debe hablar de este proceso con el administrador de seguridad.

La mayoría de los sistemas operativos basados en UNIX y Linux tienen una ubicación de sistema de archivos que contiene un conjunto 'fiable' de CA. Es posible que este sistema de archivos se haya configurado con la instalación del sistema operativo o que lo haya personalizado el administrador del sistema (por ejemplo, para incluir las CA internas que pertenecen a la organización). Las ubicaciones de estos archivos varían, pero algunos valores utilizados habitualmente para los sistemas operativos más populares son:

- 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`

Al crear y configurar un almacén de claves de IBM MQ , puede añadir fácilmente todos los archivos de certificado de un directorio, por ejemplo, `/etc/ssl/certs`, a una base de datos de claves de IBM MQ en un mandato.

Procedimiento

1. Utilice el mandato siguiente para añadir los archivos de certificado desde el directorio `/etc/ssl/certs` :

```
runmqakm -cert -add -file /etc/ssl/certs/*.pem -db mykdb.p12 -stashed
```

2. Opcional: En algunas situaciones, puede ser útil generar un conjunto 'predeterminado' de certificados para el almacén de confianza.

   Los componentes de seguridad de IBM MQ que se proporcionan con el producto proporcionan un conjunto de certificados CA 'predeterminados'.

   Nota: Es posible que estos certificados no se actualicen con frecuencia y/o tengan una vida útil relativamente corta.

   Si desea utilizar los certificados de CA preconfigurados de todos modos, puede generar un almacén de confianza utilizando los parámetros **`populate`** y **`ibmcloudtrust`** en el mandato **`runmqakm`** :

```
runmqakm -keydb -create -db mqauto.p12 -genpw -stash -type pkcs12 -populate -ibmcloudtrust
```

Conceptos relacionados

Resolución de problemas de la señal de autenticación

Tareas relacionadas

Utilización de señales de autenticación en una aplicación

Referencia relacionada

Stanza AuthToken del archivo `qm.ini`

Trabajar con certificados revocados

Las Entidades emisoras de certificados pueden revocar los certificados digitales. Puede comprobar el estado de revocación de los certificados utilizando OCSP, o listas de revocación de certificados (CRL) en servidores LDAP, dependiendo de la plataforma.

Durante el reconocimiento TLS, los participantes en la comunicación se autentican entre sí mediante certificados digitales. La autenticación puede incluir una comprobación de que el certificado recibido continúa siendo fiable. Las Entidades emisoras de certificados (CA) revocan certificados por diversas razones, entre ellas:

- El propietario ha cambiado de organización
- La clave privada ya no es secreta

Las CA publican los certificados personales revocados en una Lista de revocación de certificados (CRL). Los certificados de CA que se han revocado se publican en una Lista de revocación de autorizaciones (ARL).

En plataformas AIX, Linux, and Windows , el soporte SSL de IBM MQ comprueba si hay certificados revocados utilizando OCSP (Online Certificate Status Protocol) o utilizando CRL y ARL en servidores LDAP (Lightweight Directory Access Protocol). El OCSP es el método preferido.

IBM MQ classes for Java y IBM MQ classes for JMS no pueden utilizar la información de OCSP en un archivo de tabla de definiciones de canal de cliente. Sin embargo, puede configurar OCSP tal como se describe en Utilización de Online Certificate Protocol.

En IBM i, el soporte SSL de IBM MQ comprueba los certificados revocados utilizando CRL y ARL sólo en servidores LDAP.

En z/OS, el soporte SSL de IBM MQ comprueba los certificados revocados utilizando CRL y ARL sólo en servidores LDAP.

Para obtener más información sobre las Entidades emisoras de certificados, consulte “Certificados digitales” en la página 13.

Comprobación OCSP/CRL

La comprobación del protocolo de estado de certificados en línea (OCSP) /Lista de revocación de certificados (CRL) se realiza con respecto a los certificados entrantes remotos. El proceso comprueba toda la cadena implicada desde el certificado personal del sistema remoto hasta su certificado raíz.

Utilización de openSSL para verificar la validación de OCSP

Si su empresa utiliza openSSL para validar OCSP y, a continuación, intenta utilizar una conexión TLS de IBM Global Security Kit (GSKit) , recibirá un aviso de estado UNKNOWN.

Esto se debe a que GSKit comprueba el estado de revocación de todos los certificados de la cadena, aparte de la raíz. La operación GSKit se ajusta a RFC 5280 y esto se describe en la política de confianza de GSKit . El algoritmo GSKit intenta todos los orígenes disponibles para la información de revocación, tal como se describe en RFC 5280 y la política de confianza de GSKit .

¿Cómo funciona la comprobación OCSP/CRL en IBM MQ?

IBM MQ da soporte a dos mecanismos para controlar el comportamiento al comprobar certificados en puntos finales OCSP o CRL con nombre, ya sea en la extensión de certificado o, tal como se define en los objetos AUTHINFO:

- Los atributos **OCSPCheckExtensions**, **CDPCheckExtensions**y **OCSPAuthentication** de la stanza SSL de del archivo qm.iniy
- Utilizando el parámetro SSLCRLNL del gestor de colas y las configuraciones AUTHINFO OCSP y CRLLDAP. Consulte ALTER AUTHINFO y ALTER QMGR para obtener más información.

Atención:

El mandato ALTER AUTHINFO con **AUTHTYPE(OCSP)** no se aplica para su uso en gestores de colas IBM i o z/OS . Sin embargo, se puede especificar en esas plataformas para copiarlas en la tabla de definición de canal de cliente (CCDT) para su uso por parte del cliente.

Los atributos de stanza SSL **OCSPCheckExtensions** y **CDPCheckExtensions** controlan si IBM MQ verificará un certificado en el servidor OCSP o CRL detallado dentro de la extensión AIA del certificado.

Si no está habilitado, no se contacta con el servidor OCSP o CRL de la extensión de certificado.

Si los servidores OCSP o CRL se detallan a través de objetos AUTHINFO y se hace referencia a ellos utilizando el atributo SSLCRLNL **QMGR** , durante el proceso de revocación de certificados, IBM MQ intenta ponerse en contacto con estos servidores.

Importante: Sólo se puede definir un objeto OCSP AUTHINFO en la lista de nombres SSLCRLNL.

If:

OCSPCheckExtensions= NO y **CDPCheckExtensions**=*NO* están establecidos, y
No hay servidores OCSP o CRL definidos en objetos AUTHINFO

no se realiza ninguna comprobación de revocación de certificados.

Al verificar un certificado para su estado de revocación, IBM MQ se pone en contacto con los servidores OCSP o CRL nombrados en el orden siguiente, si está habilitado:

1. El servidor OCSP detallado en un objeto **AUTHTYPE(OCSP)** y al que se hace referencia en el atributo SSLCRLNL **QMGR** .
2. Servidores OCSP detallados en la extensión AIA de los certificados, si **OCSPCheckExtensions**=*YES*.
3. Servidores CRL detallados en la extensión **CRLDistributionPoints** de los certificados, si **CDPCheckExtensions** =*YES*.
4. Cualquier servidor CRL detallado en objetos **AUTHINFO(CRLLDAP)** y al que se hace referencia en el atributo SSLCRLNL **QMGR** .

Al verificar un certificado, si un paso hace que el servidor OCSP o el servidor CRL devuelvan una respuesta REVOKED o VALID definitiva a una consulta para el certificado, no se realizan más comprobaciones y el estado del certificado tal como se presenta se utiliza para determinar si se debe confiar en él o no.

Si un servidor OCSP o un servidor CRL devuelve un resultado de UNKNOWN, el proceso continúa hasta que un servidor OCSP o CRL devuelve un resultado definitivo o hasta que se agotan todas las opciones.

El comportamiento de si un certificado se considera revocado, si no se puede determinar su estado, es diferente para los servidores OCSP y CRL:

- Para servidores CRL, si no se puede obtener ninguna CRL, el certificado se considera NOT_REVOKED
- Para servidores OCSP, si no se puede obtener ningún estado de revocación de un servidor OCSP con nombre, el comportamiento se controla mediante el atributo **OCSPAuthentication** en la stanza SSL del archivo qm.ini .

  Puede configurar este atributo para bloquear una conexión, permitir una conexión o permitir una conexión con un mensaje de aviso.

Puede utilizar el atributo **SSLHTTPProxyName**=*string* en la stanza SSL de los archivos qm.ini y mqclient.ini para las comprobaciones de OCSP si es necesario. La serie es el nombre de host o la dirección de red del servidor proxy HTTP que GSKit debe utilizar para las comprobaciones de OCSP.

Puede establecer el valor **OCSPTimeout** en la stanza SSL de los archivos `qm.ini` o `mqclient.ini` que establece el número de segundos que se debe esperar a que un programa de respuesta OCSP realice una comprobación de revocación.

ALW Certificados revocados y OCSP

IBM MQ determina qué programa de respuesta OSCP (Online Certificate Status Protocol) se utilizará y gestiona la respuesta recibida. Puede que tenga realizar pasos para que el canal de respuesta OCSP sea accesible.

Nota: Esta información solo se aplica a IBM MQ en los sistemas AIX, Linux, and Windows.

Para comprobar el estado de revocación de un certificado digital utilizando OCSP, IBM MQ puede utilizar dos métodos para determinar el programa de respuesta OCSP con el que contactar:

- Utilizar la extensión de certificado AuthorityInfoAccess (AIA) en el certificado que se va a comprobar.
- Utilizar un URL especificado en un objeto de información de autenticación o especificado por una aplicación cliente.

Un URL especificado en un objeto de información de autenticación o mediante una aplicación cliente tiene prioridad sobre un URL en una extensión de certificados AIA.

Si el URL del programa de respuesta OCSP se oculta detrás de un cortafuegos, vuelva a configurar el cortafuegos de modo que pueda accederse al programa de respuesta OCSP o configure un servidor proxy OCSP. Especifique el nombre del servidor proxy utilizando la variable SSLHTTPProxyName en la stanza SSL. En sistemas cliente, también puede especificar el nombre del servidor proxy utilizando la variable de entorno MQSSLPROXY. Para obtener más detalles consulte la información relacionada.

Si no está preocupado si se revocan los certificados TLS, quizá porque está realizando la ejecución en un entorno de prueba, puede establecer OCSPCheckExtensions en NO en la stanza de SSL. Si establece esta variable, se hace caso omiso de la extensión de certificados AIA. No es probable que esta solución se pueda aceptar en un entorno de producción, donde probablemente no desea permitir el acceso de los usuarios que presentan certificados revocados.

La llamada para acceder al programa de respuesta OCSP puede generar uno de estos tres resultados:

Correcto
El certificado es válido.

Revocado
El certificado se revoca.

Desconocido
Esta salida se puede deber a una de las tres razones siguientes:

- IBM MQ no puede acceder al programa de respuesta OCSP.
- El programa de respuesta OCSP ha enviado una respuesta, pero IBM MQ no puede verificar la firma digital de la respuesta.
- El programa de respuesta OCSP ha enviado una respuesta que indica que no hay datos de revocación para el certificado.

Si IBM MQ recibe una salida OCSP `Desconocido`, su comportamiento depende del valor del atributo OCSPAuthentication. Para gestores de colas, este atributo se conserva en una de las ubicaciones siguientes:

- Linux AIX En la stanza SSL del archivo `qm.ini` en AIX and Linux.
- Windows En el registro de Windows.

Este atributo se puede establecer utilizando IBM MQ Explorer. Para clientes, el atributo se conserva en la stanza SSL del archivo de configuración de cliente.

Si se recibe una salida `Desconocido` y OCSPAuthentication está establecido en REQUIRED (el valor predeterminado), IBM MQ rechaza la conexión y emite un mensaje de error del tipo AMQ9716. Si están habilitados mensajes de sucesos SSL del gestor de colas, se genera un mensaje de suceso SSL del tipo MQRC_CHANNEL_SSL_ERROR con ReasonQualifier establecido en MQRQ_SSL_HANDSHAKE_ERROR.

Si se recibe una salida `Desconocido` y OCSPAuthentication está establecido en OPTIONAL, IBM MQ permite que se inicie el canal SSL y no se genere ningún aviso o mensajes de suceso SSL.

Si se recibe una salida `Desconocido` y OCSPAuthentication está establecido en WARN, se inicia el canal SSL pero IBM MQ emite un mensaje de aviso del tipo AMQ9717 en el registro de errores. Si están habilitados los mensajes de sucesos SSL, se genera un mensaje de sucesos SSL del tipo MQRC_CHANNEL_SSL_WARNING con ReasonQualifier establecido en MQRQ_SSL_UNKNOWN_REVOCATION.

Firma digital de respuestas OCSP

Un programa de respuesta OCSP puede firmar sus respuestas de una de tres formas. El programa de respuesta le informará del método que se utiliza.

- El programa de respuesta OCSP puede firmarse digitalmente utilizando el mismo certificado CA que emitió el certificado que está comprobando. En este caso, no es necesario que configure ningún certificado adicional; los pasos que haya realizado para establecer la conectividad TLS serán suficientes para verificar la respuesta OCSP.
- La respuesta OCSP se puede firmar de forma digital utilizando otro certificado firmado por la misma entidad emisora de certificados (CA) que emitió el certificado que se está comprobando. El certificado de firma se envía junto con la respuesta OCSP en este caso. El certificado transmitido del programa de respuesta OCSP debe tener una extensión de uso de claves ampliado establecida en `id-kp-OCSPSigning` para que sea fiable para este fin. Debido a que la respuesta OCSP se envía con el certificado que la firmó (y dicho certificado está firmado por una CA que ya es fiable para la conectividad TLS) no es necesaria ninguna configuración adicional del certificado.
- La respuesta OCSP se puede firmar digitalmente utilizando otro certificado que no esté relacionado directamente con el certificado que está comprobando. En este caso, la respuesta OCSP está firmada por un certificado emitido por el propio programa de respuesta OCSP. Debe añadir una copia del certificado de respondedor OCSP a la base de datos de claves del cliente o gestor de colas que realiza la comprobación OCSP. Consulte “Adición de un certificado de CA, o la parte pública de un certificado de confianza, a un repositorio de claves en AIX, Linux, and Windows” en la página 560. Cuando se añade un certificado CA, de forma predeterminada se añade como raíz fiable, que es el valor necesario en este contexto. Si este certificado no se añade, IBM MQ no puede verificar la firma digital en la respuesta de OCSP y la comprobación de OCSP tiene como resultado una salida Desconocido, lo que podría hacer que IBM MQ cerrara el canal, en función del valor de OCSPAuthentication.

OCSP (Online Certificate Status Protocol) en aplicaciones cliente de Java y JMS.

Debido a una limitación de la API de Java, IBM MQ puede utilizar la comprobación de revocación de certificados OCSP (Online Certificate Status Protocol) para sockets seguros TLS únicamente cuando se habilita OCSP para todo el proceso de la máquina virtual Java (JVM). Hay dos modos de habilitar OCSP para todos los sockets seguros de la JVM:

- Editar el archivo JRE java.security para incluir los valores de configuración de OCSP que se muestran en la Tabla 1 y reiniciar la aplicación.
- Utilizar la API java.security.Security.setProperty(), sujeta a cualquier política de Java Security Manager que esté en vigor.

Como mínimo, debe especificar uno de los valores ocsp.enable y ocsp.responderURL.

Nombre de propiedad	Descripción
ocsp.enable	El valor de esta propiedad es `true` o `false`. Si es `true`, se habilita la comprobación OCSP cuando se lleva a cabo la comprobación de revocación de certificados. Si el valor es `false` no está establecido, la comprobación OCSP está inhabilitada.
ocsp.responderURL	El valor de esta propiedad es un URL que identifica la ubicación del programa de respuesta OCSP. El siguiente es un ejemplo: `ocsp.responderURL=http://ocsp.example.net:80`. De forma predeterminada, la ubicación del programa de respuesta OCSP se determina de forma implícita a partir del certificado que se está validando. La propiedad se utiliza cuando en el certificado falta la extensión de Authority Information Access (definida en RFC 3280) o cuando requiere una alteración temporal.
ocsp.responderCertSubjectName	El valor de esta propiedad es el nombre del asunto del certificado del programa de respuesta OCSP. El siguiente es un ejemplo: `ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"`. De forma predeterminada, el certificado del programa de respuesta OSCP es el del emisor del certificado que se está validando. Esta propiedad identifica el certificado del programa de respuesta OSCP cuando el valor predeterminado no se aplica. Su valor es una serie de nombre distinguido (definido en RFC 2253) que identifica un certificado en el conjunto de certificados que se suministran durante la validación de la vía de acceso de certificado. En los casos en los que el nombre del asunto no es suficiente para identificar de forma exclusiva el certificado, en su lugar, se deben utilizar las dos propiedades ocsp.responderCertIssuerName y ocsp.responderCertSerialNumber. Cuando se establece esta propiedad, se omiten las propiedades ocsp.responderCertIssuerName y ocsp.responderCertSerialNumber.
ocsp.responderCertIssuerName	El valor de esta propiedad es el nombre del emisor del certificado del programa de respuesta OCSP. El siguiente es un ejemplo: `ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"`. De forma predeterminada, el certificado del programa de respuesta OSCP es el del emisor del certificado que se está validando. Esta propiedad identifica el certificado del programa de respuesta OSCP cuando el valor predeterminado no se aplica. Su valor es una serie de nombre distinguido (definido en RFC 2253) que identifica un certificado en el conjunto de certificados que se suministran durante la validación de la vía de acceso de certificado. Cuando se establece esta propiedad, también se debe establecer la propiedad ocsp.responderCertSerialNumber. Esta propiedad se omite cuando se establece la propiedad ocsp.responderCertSubjectName.
ocsp.responderCertSerialNumber	El valor de esta propiedad es el número de serie del certificado del programa de respuesta OCSP. El siguiente es un ejemplo: `ocsp.responderCertSerialNumber=2A:FF:00`. De forma predeterminada, el certificado del programa de respuesta OSCP es el del emisor del certificado que se está validando. Esta propiedad identifica el certificado del programa de respuesta OSCP cuando el valor predeterminado no se aplica. Este valor es una serie de dígitos hexadecimales (pueden haber separadores de espacio o de signo de dos puntos) que identifica un certificado en el conjunto de certificados que se suministran durante la validación de la vía de acceso de certificado. Cuando se establece esta propiedad, también

Nombre de propiedad	Descripción
	se debe establecer la propiedad ocsp.responderCertIssuerName. Esta propiedad se omite cuando se establece la propiedad ocsp.responderCertSubjectName.

Antes de habilitar OCSP de este modo, existen varios puntos a tener en cuenta:

- Cuando se establece configuración de OCSP, todos los sockets seguros del proceso de la JVM resultan afectados. En algunos casos, es posible que esta configuración tenga efectos colaterales no deseados cuando la JVM se comparte con otro código de la aplicación que utiliza los sockets seguros TLS. Asegúrese de que la configuración OCSP elegida sea adecuada para todas las aplicaciones que se ejecutan en la misma JVM.
- Cuando se aplica el mantenimiento a JRE es posible que se sobrescriba el archivo java.security. Preste atención cuando aplique el mantenimiento del producto y los arreglos temporales de Java para no sobrescribir el archivo java.security. Es posible que sea necesario volver a aplicar los cambios de java.security después de aplicar el mantenimiento. Por este motivo, en su lugar, puede definir la configuración de OCSP mediante la API java.security.Security.setProperty().
- Cuando se habilita la comprobación OCSP, ésta solo tiene efecto si también está habilitada la comprobación de revocación. La comprobación de revocación se habilita mediante el método `PKIXParameters.setRevocationEnabled()`.
- Si está utilizando el interceptor AMS de Java que se describe en la sección Habilitación de la comprobación OCSP en los interceptores, preste atención y evite utilizar una configuración de java.security de OCSP que entre en conflicto con la configuración OCSP de AMS en el archivo de configuración del almacén de claves.

Trabajar con listas de revocación de certificados y listas de revocación de autorizaciones

El soporte de IBM MQ para las CRL y las ARL varía según la plataforma.

El soporte de CRL y ARL en cada plataforma es el siguiente:

- Multi En Multiplatforms, el soporte de CRL y ARL cumple con las recomendaciones del perfil de CRL PKIX X.509 V2 .
- z/OS En z/OS, SSL del sistema da soporte a las CRL y las ARL almacenadas en servidores LDAP por el producto Tivoli Public Key Infrastructure.

IBM MQ mantiene una memoria caché de las CRL y las ARL a las que se ha accedido en las últimas 12 horas.

Cuando un gestor de colas o un cliente IBM MQ MQI client recibe un certificado, comprueba la CRL para confirmar que el certificado sigue siendo válido. IBM MQ comprueba en primer lugar la memoria caché, si ésta existe. Si la CRL no está en la memoria caché, IBM MQ interroga a las ubicaciones del servidor CRL de LDAP en el orden en que aparecen en la lista de nombres de los objetos de información de autenticación especificados por el atributo *SSLCRLNL* , hasta que IBM MQ encuentre una CRL disponible. Si no se especifica la lista de nombres o si se especifica con un valor en blanco, las CRL no se comprueban.

Configuración de los servidores LDAP

Configure la estructura de Árbol de información de directorios de LDAP para que refleje la jerarquía de Nombres distinguidos de las CA. Para ello, utilice archivos de Formato de intercambio de datos LDAP (LDIF).

Configure la estructura de Árbol de información de directorios (DIT) de LDAP, de modo que utilice la jerarquía correspondiente a los nombres distinguidos de las CA que emiten los certificados y las CRL. Puede configurar la estructura DIT con un archivo que utilice el Formato de intercambio de datos LDAP (LDIF). También puede utilizar archivos LDIF para actualizar un directorio.

Los archivos LDIF son archivos de texto ASCII que contienen la información necesaria para definir objetos en un directorio LDAP. Los archivos LDIF contienen una o varias entradas, cada una de las cuales consta de un Nombre distinguido, como mínimo una definición de clase de objeto y, opcionalmente, varias definiciones de atributo.

El atributo `certificateRevocationList;binary` contiene una lista, con formato binario, de los certificados de usuario revocados. El atributo `authorityRevocationList;binary` contiene una lista con formato binario de certificados de CA revocados. Para la utilización con IBM MQ TLS, los datos binarios para estos atributos deben cumplir con el formato DER (Definite Encoding Rules). Para obtener más información acerca de los archivos LDIF, consulte la documentación que se proporciona con el servidor LDAP.

Figura 20 en la página 352 muestra un archivo LDIF de ejemplo que puede crear como entrada al servidor LDAP para cargar los CRL y ARL emitidos por CA1, que es una entidad emisora de certificados imaginaria con el nombre distinguido "CN=CA1, OU=Test, O=IBM, C=GB", configurado por la organización de prueba en 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. Archivo LDIF de ejemplo para una Entidad emisora de certificados. Puede variar de implementación en implementación.

La Figura 21 en la página 352 muestra la estructura DIT que el servidor LDAP crea cuando carga el archivo LDIF de ejemplo que se muestra en la Figura 20 en la página 352 junto con un archivo similar para la CA2, una Entidad emisora de certificados ficticia establecida por la organización PKI, también dentro de IBM.

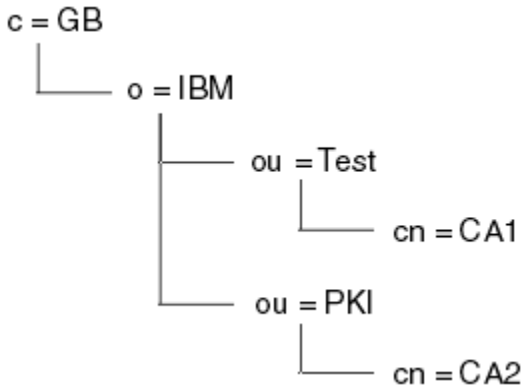


Figura 21. Ejemplo de una estructura de árbol de la información de directorios LDAP

IBM MQ comprueba las CRL y las ARL.

Nota: Asegúrese de que la lista de control de accesos de su servidor LDAP permita que los usuarios autorizados lean, busquen y comparen las entradas que contienen las CRL y las ARL. IBM MQ accede al servidor LDAP utilizando las propiedades LDAPUSER y LDAPPWD del objeto AUTHINFO.

Configuración y actualización de los servidores LDAP
Utilice este procedimiento para configurar o actualizar el servidor LDAP.

1. Obtenga las CRL y ARL en formato DER de su autoridad o autoridades de certificación.
2. Con un editor de texto o la herramienta que le proporcione el servidor LDAP, cree uno o varios archivos LDIF que contengan el nombre distinguido de la CA y las definiciones de clases de objetos necesarias. Copie los datos con formato DER en el archivo LDIF como valores del atributo `certificateRevocationList;binary` para las CRL, del atributo `authorityRevocationList;binary` para las ARL, o ambos.
3. Inicie el servidor LDAP.
4. Añada las entradas del archivo o archivos LDIF que ha creado en el paso "2" en la página 353.

Cuando haya configurado el servidor CRL LDAP, compruebe que se ha configurado correctamente. Primero, intente utilizar un certificado que no se haya revocado en el canal, y compruebe que el canal se inicia correctamente. A continuación, utilice un certificado que se haya revocado y compruebe que el canal no se inicia correctamente.

Obtenga las CRL actualizadas de las Autoridades de certificación de forma regular. Se recomienda que lo haga en sus servidores LDAP cada 12 horas.

Acceso a las CRL y las ARL con un gestor de colas

Un gestor de colas está asociado a uno o más objetos de información de autenticación, que contienen la dirección de un servidor CRL LDAP. IBM i IBM MQ en IBM i se comporta de forma diferente a otras plataformas.

Tenga en cuenta que en este apartado, la información sobre las listas de revocación de certificados (CRL) también es aplicable para las listas de revocación de autorizaciones (ARL).

Debe indicar al gestor de colas cómo acceder a las CRL proporcionándole objetos de información de autenticación, cada uno de los cuales contiene la dirección de un servidor CRL LDAP. Los objetos de información de autenticación se mantienen en una lista de nombres, que se especifica en el atributo de gestor de colas *SSLCRLNL* .

En el ejemplo siguiente, MQSC se utiliza para especificar los parámetros:

1. Defina los objetos de información de autenticación con el mandato MQSC, DEFINE AUTHINFO, con el parámetro AUTHTYPE establecido en CRLLDAP. IBM i En IBM i, también puede utilizar el mandato CL CRTMQMAUTI.

   El valor CRLLDAP para el parámetro AUTHTYPE indica que se accede a las CRL en servidores LDAP. Cada objeto de información de autenticación con el tipo CRLLDAP que cree contendrá la dirección de un servidor LDAP. Cuando tenga más de un objeto de información de autenticación, los servidores LDAP a los que apuntan deben contener información idéntica. Esto permite que el servicio continúe si uno o varios servidores LDAP no se ejecutan correctamente.

   z/OS Asimismo, únicamente en z/OS, se debe acceder a todos los servidores LDAP utilizando el mismo ID de usuario y contraseña. El ID de usuario y la contraseña utilizados son los especificados en el primer objeto AUTHINFO de la lista de nombres.

   En todas las plataformas, el ID de usuario y la contraseña se envían al servidor LDAP sin cifrar.
2. Con el mandato MQSC, DEFINE NAMELIST, defina una lista de nombres para los nombres de los objetos de información de autenticación. z/OS En z/OS, asegúrese de que el atributo de lista de nombres NLTYPE esté establecido en AUTHINFO.
3. Con el mandato MQSC, ALTER QMGR, proporcione la lista de nombres al gestor de colas. Por ejemplo:

   ```
   ALTER QMGR SSLCRLNL(sslcrlnlname)
   ```

   donde `sslcrlnlname` es la lista de nombres de los objetos de información de autenticación.

Este mandato establece un atributo de gestor de colas denominado *SSLCRLNL*. El valor inicial del gestor de colas para este atributo está en blanco.

IBM i En IBM i, puede especificar objetos de información de autenticación, pero el gestor de colas no utiliza ni objetos de información de autenticación ni una lista de nombres de objetos de información de autenticación. Solamente los clientes IBM MQ que utilizan una tabla de conexiones de cliente generada por un gestor de colas de IBM i utilizan la información de autenticación especificada para dicho gestor de colas de IBM i. El atributo de gestor de colas *SSLCRLNL* en IBM i determina qué información de autenticación utilizan estos clientes. Consulte "Acceso a las CRL y las ARL en IBM i" en la página 354 para obtener información sobre cómo indicar a un gestor de colas de IBM i cómo acceder a las CRL.

Puede añadir hasta 10 conexiones a servidores LDAP alternativos a la lista de nombres, para asegurarse de la continuidad del servicio si uno o varios servidores LDAP no respondieran. Tenga en cuenta que los servidores LDAP deben contener información idéntica.

IBM i *Acceso a las CRL y las ARL en IBM i*

Utilice este procedimiento para acceder a las CRL o las ARL en IBM i.

Tenga en cuenta que en este apartado, la información sobre las listas de revocación de certificados (CRL) también es aplicable para las listas de revocación de autorizaciones (ARL).

Siga estos pasos para configurar una ubicación CRL para un certificado específico en IBM i:

1. Acceda a la interfaz DCM, tal y como se describe en el apartado "Acceso a DCM" en la página 281.
2. En la categoría de tareas **Gestionar ubicaciones CRL** del panel de navegación, pulse **Añadir ubicación CRL**. Se visualiza la página Ubicaciones CRL en la sección de tareas.
3. En el campo **Nombre de ubicación de CRL** , escriba un nombre de ubicación de CRL, por ejemplo `LDAP Server #1`
4. En el campo **Servidor LDAP**, escriba el nombre de servidor LDAP.
5. En el campo **Utilizar Secure Sockets Layer (SSL)**, seleccione **Sí** si desea conectarse al servidor LDAP utilizando TLS. De lo contrario, seleccione **No**.
6. En el campo **Número de puerto**, escriba un número de puerto para el servidor LDAP; por ejemplo, 389.
7. Si el servidor LDAP no permite que los usuarios anónimos consulte el directorio, escriba un nombre distinguido de inicio de sesión para el servidor en el campo **nombre distinguido de inicio de sesión**.
8. Pulse **Aceptar**. DCM le informa de que ha creado la ubicación CRL.
9. En el panel de navegación, pulse **Seleccionar un almacén de certificados**. La página Seleccionar un almacén de certificados se visualiza en la sección de tareas.
10. Seleccione el recuadro **Otro almacén de certificados del sistema** y pulse **Continuar**. Se visualiza la página Almacén de certificados y contraseña.
11. En el campo **Vía de acceso y nombre de archivo del almacén de certificados**, escriba la vía de acceso y nombre de archivo de IFS que estableció en el apartado "Crear un almacén de certificados en IBM i" en la página 284.
12. Escriba una contraseña en el campo **Contraseña del almacén de certificados**. Pulse **Continuar**. La página Almacén de certificados actual se visualiza en la sección de tareas.
13. En la categoría de tareas **Gestionar certificados** del panel de navegación, pulse **Actualizar asignación de ubicación CRL**. La página Asignación de ubicación CRL se visualiza en la sección de tareas.
14. Seleccione el botón correspondiente al certificado de CA al que desea asignar la ubicación CRL. Pulse **Actualizar asignación de ubicación CRL**. La página Actualizar asignación de ubicación CRL se visualiza en la sección de tareas.
15. Seleccione el botón para la ubicación CRL que desea asignar al certificado. Pulse **Actualizar asignación**. DCM le informa de que ha actualizado la asignación.

Tenga en cuenta que DCM le permite asignar un servidor LDAP diferente mediante la Entidad emisora de certificados.

Acceso a las CRL y las ARL utilizando IBM MQ Explorer
Puede utilizar IBM MQ Explorer para indicar a un gestor de colas cómo acceder a las CRL.

Tenga en cuenta que en este apartado, la información sobre las listas de revocación de certificados (CRL) también es aplicable para las listas de revocación de autorizaciones (ARL).

Utilice el procedimiento siguiente para establecer una conexión LDAP con una CRL:

1. Asegúrese de que ha iniciado el gestor de colas.
2. Pulse el botón derecho del ratón en la carpeta **Información de autenticación** y pulse **Nuevo -> Información de autenticación**. En la hoja de propiedades que se abrirá:
   a. En la primera página **Crear información de autenticación**, escriba un nombre para el objeto CRL(LDAP).
   b. En la página **General** de **Modificar las propiedades**, seleccione el tipo de conexión. De manera opcional, puede escribir una descripción.
   c. Seleccione la página **CRL(LDAP)** de **Modificar las propiedades**.
   d. Escriba el nombre del servidor LDAP como el nombre de red o la dirección IP.
   e. Si el servidor requiere detalles para la conexión, proporcione un ID de usuario y, si es necesario, una contraseña.
   f. Pulse **Aceptar**.
3. Pulse el botón derecho del ratón en la carpeta Listas de nombres y pulse **Nuevo-> Lista de nombres**. En la hoja de propiedades que se abrirá:
   a. Escriba un nombre para la lista de nombres.
   b. Añada a la lista el nombre del objeto CRL(LDAP) (del paso "2.a" en la página 355).
   c. Pulse **Aceptar**.
4. Pulse el botón derecho del ratón en el gestor de colas, seleccione **Propiedades** y seleccione la página **SSL**:
   a. Seleccione el recuadro de selección **Comprobar los certificados enviados a este gestor de colas respecto a las listas de revocación de certificados**.
   b. Escriba el nombre de la lista de nombres (del paso "3.a" en la página 355) en el campo **Nombre de la lista de CRL**.

Acceso a las CRL y las ARL con un IBM MQ MQI client

Existen tres formas de especificar los servidores LDAP que contienen listas de revocación de certificados (CRL) para su comprobación por parte de un IBM MQ MQI client.

Tenga en cuenta que en este apartado, la información sobre las listas de revocación de certificados (CRL) también es aplicable para las listas de revocación de autorizaciones (ARL).

A continuación se indican las tres formas de especificar los servidores LDAP:

- Utilizar una tabla de definiciones de canal
- Utilizar la estructura de opciones de configuración de SSL, MQSCO, en una llamada MQCONNX
- Utilizar Active Directory (en sistemas Windows con soporte para Active Directory)

Para más detalles, consulte la información relacionada.

Puede incluir hasta 10 conexiones a servidores LDAP alternativos para asegurarse de la continuidad del servicio si uno o más servidores LDAP no se realiza correctamente n. Tenga en cuenta que los servidores LDAP deben contener información idéntica.

No puede acceder a las CRL de LDAP desde un canal de cliente MQI de IBM MQ MQI client que se ejecuta en Linux (plataforma zSeries).

Ubicación de un programa de respuestas OCSP, y de servidores LDAP que contienen CRL
En un sistema de IBM MQ MQI client, puede especificar la ubicación de un programa de respuestas OCSP y de servidores LDAP (Lightweight Directory Access Protocol) que tienen las CRL (listas de revocación de certificados).

Puede especificar estas ubicaciones de tres formas, se describen aquí en orden de mayor a menor prioridad.

IBM i Para IBM i, consulte Acceso a las CRL y las ARL en IBM i.

Cuando una aplicación cliente IBM MQ MQI client emite una llamada MQCONNX

Puede especificar una respuesta OCSP o un servidor LDAP que contiene CRL en una llamada **MQCONNX**.

En una llamada **MQCONNX**, la estructura de opciones de conexión, MQCNO, puede hacer referencia a una estructura de opciones de configuración SSL, MQSCO. A su vez, la estructura MQSCO puede hacer referencia a una o varias estructuras de registro de información de autenticación, MQAIR. Cada estructura MQAIR contiene toda la información que un IBM MQ MQI client necesita para acceder a una respuesta OCSP o un servidor LDAP que contiene CRL. Por ejemplo, uno de los campos de una estructura MQAIR es el URL en el que se puede contactar con una respuesta. Para obtener más información acerca de la estructura MQAIR, consulte MQAIR - Registro de información de autenticación.

Utilización de una tabla de definiciones de canal de cliente (CCDT) para acceder a un programa de respuestas OCSP o a servidores LDAP

Para que un IBM MQ MQI client pueda acceder a un programa de respuestas OCSP o a servidores LDAP que contienen CRL, incluya los atributos de uno o más objetos de información de autenticación en una tabla de definiciones de canal de cliente.

En un gestor de colas de servidor, puede definir uno o varios objetos de información de autenticación. Los atributos de un objeto de autenticación contienen toda la información necesaria para acceder a un programa de respuestas OCSP (en plataformas donde se admite OCSP) o a un servidor LDAP que contiene CRL. Uno de los atributos especifica el URL del programa de respuestas OCSP, el otro especifica la dirección de host, o la dirección IP, de un sistema en que se ejecuta un servidor LDAP.

z/OS IBM i Un objeto de información de autenticación con AUTHTYPE(OCSP) no es aplicable para utilizarse en gestores de colas de IBM i o z/OS, pero puede especificarse en las plataformas que deben copiarse a la tabla de definiciones del canal de cliente (CCDT) para que las use el cliente.

Para que un cliente MQI de IBM MQ MQI client pueda acceder a un programa de respuestas OCSP o a servidores LDAP que contienen CRL, pueden incluirse los atributos de uno o más objetos de información de autenticación en una tabla de definiciones de canal de cliente. Puede incluir dichos atributos de una de las maneras siguientes:

Multi

En plataformas de servidor AIX, Linux, IBM i y Windows

Puede definir una lista de nombres que contenga los nombres de uno o varios objetos de información de autenticación. A continuación, puede establecer el atributo del gestor de colas, **SSLCRLNL**, en el nombre de esta lista de nombres.

Si utiliza CRL, puede configurarse más de un servidor LDAP para ofrecer más disponibilidad. La intención es que cada servidor LDAP contenga las mismas CRL. Si un servidor LDAP no está disponible cuando se necesita, un IBM MQ MQI client puede intentar acceder a otro.

Los atributos de los objetos de información de autenticación identificados por la lista de nombres se denominan colectivamente *ubicación de la revocación del certificado*. Cuando establece el atributo de gestor de colas, **SSLCRLNL**, en el nombre de la lista de nombres, la ubicación de revocación de certificados se copia en la tabla de definiciones de canal de cliente asociada con el gestor de colas. Si puede accederse a la CCDT desde un sistema cliente como archivo compartido, o si la CCDT se

copia posteriormente en un sistema de cliente, el IBM MQ MQI client de dicho sistema puede utilizar la ubicación de revocación de certificados de la CCDT para acceder a un programa de respuesta OCSP o a servidores LDAP que contienen CRL.

Si la ubicación de la revocación del certificado del gestor de colas se cambia posteriormente, el cambio se refleja en la CCDT asociada al gestor de colas. Si el atributo de gestor de colas, **SSLCRLNL**, se establece en blanco, la ubicación de revocación de certificado se elimina de la CCDT. Estos cambios no quedan reflejados en ninguna copia de la tabla en un sistema cliente.

Si necesita que la ubicación de revocación del certificado en los extremos del cliente y del servidor de un canal MQI sea diferente, y ha utilizado el gestor de colas del servidor para crear la información de la ubicación de revocación del certificado, puede hacer lo siguiente:

1. En el gestor de colas del servidor, cree la información de la ubicación de revocación del certificado que se utilizará en el sistema cliente.
2. Copie la CCDT que contiene la ubicación de revocación del certificado al sistema de cliente.
3. En el gestor de colas del servidor, cambie la ubicación de revocación del certificado por la que se necesita en el extremo del servidor del canal MQI.
4. En la máquina de cliente, puede utilizar el mandato **runmqsc** con el parámetro **-n**.

Multi

En plataformas cliente AIX, Linux, IBM i y Windows

Puede crear una CCDT en la máquina de cliente mediante el mandato runmqsc con el parámetro **-n** y los objetos **DEFINE AUTHINFO** en el archivo CCDT. El orden en que se definen los objetos será el orden en que se utilizan en el archivo. Cualquier nombre que utilice en un objeto **DEFINE AUTHINFO** no se retendrá en el archivo. Solo se utilizan números de posición cuando ejecute **DISPLAY** para los objetos **AUTHINFO** de un archivo CDDT.

Nota: Si especifica el parámetro **-n**, no debe especificar ningún otro parámetro.

Utilización de Active Directory en Windows

Windows

En sistemas Windows, puede utilizar el mandato de control **setmqcrl** para publicar la información de CRL actual en Active Directory.

El mandato **setmqcrl** no publica información OCSP.

Para obtener información sobre este mandato y su sintaxis, consulte setmqcrl.

Acceso a las CRL y las ARL con un IBM MQ classes for Java y IBM MQ classes for JMS

IBM MQ classes for Java y IBM MQ classes for JMS acceden a las CRL de forma diferente de otras plataformas.

Para obtener información sobre cómo trabajar con CRL y ARL con IBM MQ classes for Java, consulte Utilización de listas de revocación de certificados

Si desea más información sobre cómo trabajar con CRL y ARL con IBM MQ classes for JMS, consulte Propiedad de objeto SSLCERTSTORES

Manipulación de objetos de información de autenticación

Puede manipular objetos de información de autenticación utilizando mandatos MQSC o PCF, o mediante IBM MQ Explorer.

Los mandatos MQSC siguientes actúan en los objetos de información de autenticación:

- DEFINE AUTHINFO
- ALTER AUTHINFO
- DELETE AUTHINFO

- DISPLAY AUTHINFO

Si desea una descripción completa de estos mandatos, consulte Mandatos MQSC.

Los mandatos PCF (Programmable Command Format) siguientes actúan en los objetos de información de autenticación:

- Crear información de autenticación
- Copiar información de autenticación
- Modificar información de autorización
- Suprimir información de autenticación
- Consultar información de autenticación
- Consultar nombres de información de autenticación

Si desea una descripción completa de estos mandatos, consulte Definiciones de los formatos de mandato programables.

En plataformas donde esté disponible, también puede utilizar IBM MQ Explorer.

Linux AIX Utilización de PAM (Pluggable Authentication Method)

Puede utilizar PAM únicamente en las plataformas AIX and Linux. Un sistema AIX o Linux típico tiene módulos PAM que implementan el mecanismo de autenticación tradicional; sin embargo, puede haber más. Del mismo modo que la tarea básica de validación de contraseñas, los módulos PAM también pueden invocarse para llevar a cabo reglas adicionales.

Los archivos de configuración definen qué método de autenticación se va a utilizar para cada aplicación. Las aplicaciones de ejemplo incluyen el inicio de sesión de terminal estándar, ftp y telnet.

La ventaja de PAM es que la aplicación no necesita saber, o preocuparse, de cómo se autentica en realidad el ID de usuario. Siempre que la aplicación pueda proporcionar una forma correcta de datos de autenticación para PAM, el mecanismo detrás es transparente.

El formato de los datos de autenticación depende del sistema que se utiliza. Por ejemplo, IBM MQ obtiene una contraseña a través de parámetros, como por ejemplo la estructura MQCSP utilizada en la llamada de API MQCONNX.

Importante: No puede establecer el atributo **AUTHENMD** hasta que instale IBM MQ 8.0.0 Fix Pack 3y, a continuación, reinicie el gestor de colas, utilizando un **-e CMDLEVEL=***nivel* de *802* (en el mandato strmqm) para establecer el nivel de mandatos que necesita.

Configuración del sistema para utilizar PAM

El nombre de servicio utilizado por IBM MQ, al invocar PAM, es *ibmmq*.

Tenga en cuenta que una instalación de IBM MQ intenta mantener una configuración PAM predeterminada que permite conexiones para los usuarios del sistema operativo, basándose en valores predeterminados conocidos para los distintos sistemas operativos.

Sin embargo, el administrador del sistema debe verificar qué reglas definidas en los archivos `/etc/pam.conf` o `/etc/pam.d/ibmmq` siguen siendo apropiadas.

Autorización del acceso a objetos

Esta sección contiene información sobre cómo utilizar el gestor de autorizaciones sobre objetos y programas de salida de canal para controlar el acceso a los objetos.

ALW En sistemas AIX, Linux, and Windows . el acceso a los objetos se controla utilizando el gestor de autorizaciones sobre objetos (OAM). Esta colección de temas contiene información sobre cómo utilizar la interfaz de mandatos en el OAM.

Esta sección también contiene una lista de comprobación que puede utilizar para determinar qué tareas realizar para aplicar seguridad al sistema en todas las plataformas, y las consideraciones para otorgar a usuarios la autorización para administrar IBM MQ y para trabajar con objetos IBM MQ.

Si los mecanismos de seguridad proporcionados no satisfacen sus necesidades, puede desarrollar sus propios programas de salida de canal.

Determinar qué usuario se utiliza para la autorización

Las autorizaciones para acceder a los recursos se otorgan a los grupos de los que el usuario es miembro o, en determinadas modalidades, directamente al usuario asociado a la conexión. Durante el proceso de conexión, y en particular para las conexiones remotas (cliente), la configuración del gestor de colas podría cambiar esta identidad. Esta página lista las distintas características de IBM MQ y sus opciones de configuración que podrían afectar a la identidad de una aplicación de conexión y al orden de prioridad en el que estas características entran en vigor.

Características que pueden modificar qué usuario se adopta

Las distintas características que pueden establecer qué usuario debe estar autorizado son las siguientes:

Usuario confirmado por aplicación
Cuando IBM MQinicia una conexión remota, el usuario del sistema operativo con el que se ejecuta el proceso se envía al gestor de colas receptor. Este usuario se envía para asegurarse de que si no existe ninguna configuración adicional que modifique el usuario, hay un usuario que se puede utilizar para la comprobación de autorización.

No se recomienda utilizar este usuario como base para la autorización, ya que permite a las conexiones confirmar su identidad sin ninguna validación del lado del servidor. Esto puede incluir incluso al usuario administrativo ('mqm ').

Valor MCAUSER de canal
Las aplicaciones que se conectan a través de enlaces de red lo hacen utilizando una definición de canal de IBM MQ . Las definiciones de canal dan soporte al atributo **MCAUSER** , que se puede utilizar para especificar un usuario diferente que se utilizará para la autorización en lugar del usuario confirmado por las aplicaciones de conexión.

Autenticación de conexión ADOPTCTX
Las aplicaciones pueden especificar un usuario y una contraseña para enviarlos a un gestor de colas con fines de autenticación. Estas credenciales se autentican utilizando la configuración especificada para la característica de autenticación de conexión. La opción **ADOPTCTX** para la autenticación de conexión controla si se debe utilizar un usuario para la autorización después de que se haya validado correctamente. Si se establece en YES, el usuario que se proporciona para la autenticación se adopta para las comprobaciones de autorización.

V 9.4.0 A partir de IBM MQ 9.3.4, se puede proporcionar una señal para la autenticación, si **ADOPTCTX** se establece en YES, se adopta un usuario a partir de las reclamaciones que contiene la señal.

Registro de autenticación de canal MCAUSER
Durante el proceso de conexión, el gestor de colas intentará encontrar un registro de autenticación de canal que coincida con la conexión. Si un registro de autenticación de canal coincide y su valor de atributo **USERSRC** se establece en MAP, IBM MQ cambia el usuario utilizado para las autorizaciones al valor del atributo **MCAUSER** .

Salidas de seguridad
Las salidas de seguridad son funciones personalizadas que se pueden escribir y llamar durante el proceso de seguridad de IBM MQ . Cuando se llama a la función, se proporciona con una copia de la estructura MQCD que incluye varios campos relacionados con el usuario de conexiones que se utilizará para las comprobaciones de autorización. Las salidas de seguridad pueden modificar estos campos para cambiar el usuario que se autorizará.

orden de prioridad

La tabla siguiente muestra el orden de prioridad para cada característica de seguridad descrita en “Características que pueden modificar qué usuario se adopta” en la página 359 cuando IBM MQ selecciona un usuario para autorizar. El orden es de menor a mayor, es decir, una característica de seguridad que establece un usuario en la primera fila se altera temporalmente por cualquiera de las otras filas.

Tabla 71. Orden de prioridad para las características de seguridad

Orden	**Característica**
1 (más bajo)	ID confirmado de aplicación
2	Atributo **MCAUSER** de definición de canal
3	Autenticación de conexión con **ADOPTCTX(YES)**
4	Registros de autenticación de canal con **USERSRC(MAP)**
5 (más alto)	Salida de seguridad

Implicaciones de la adopción temprana

Los registros de autenticación de conexión y autenticación de canal proporcionan una opción de configuración que controla cuándo se realiza la adopción del usuario de autenticación de conexión. Este valor se conoce como adopción temprana. Si la adopción temprana está habilitada, la adopción de identidad de autenticación de conexión se produce antes de que se procesen los registros de autenticación de canal (lo que significa que los registros de autenticación de canal alteran temporalmente cualquier adopción de **CONNAUTH**).

Si está inhabilitado, el orden se invierte, es decir, los registros de autenticación de canal se procesan antes de la adopción de **CONNAUTH** . En esta situación, la adopción de autenticación de conexión tiene una prioridad efectiva más alta que los registros de autenticación de canal.

El valor predeterminado para la adopción temprana es `enabled`.

ALW Control del acceso a objetos mediante el OAM en AIX, Linux, and Windows

El Gestor de autorizaciones sobre objetos (OAM) proporciona una interfaz de mandatos para otorgar y revocar autorización a objetos IBM MQ.

Debe tener la autorización adecuada para utilizar estos mandatos, como se describe en “Autorización para administrar IBM MQ en AIX, Linux, and Windows” en la página 408. Los ID de usuario que están autorizados para administrar IBM MQ tiene autorización de *superusuario* para el gestor de colas, lo que significa que no tiene que otorgarles más permisos para emitir mandatos o solicitudes MQI.

Linux AIX Permisos basados en usuario de OAM en AIX and Linux

En sistemas UNIX and Linux , el gestor de autorizaciones sobre objetos (OAM) puede utilizar la autorización basada en usuario, así como la autorización basada en grupo.

Antes de IBM MQ 8.0, las listas de control de accesos (ACL) en UNIX and Linux solo se basan en grupos. DeIBM MQ 8.0 , las ACL se basan tanto en ID de usuario como en grupos, y puede usar el modelo basado en usuarios o el modelo basado en grupos para la autorización configurando el**SecurityPolicy** atribuir al valor apropiado como se describe en Estrofa de servicio delqm.ini archivo .

Cambios en el comportamiento para IBM MQ 8.0 y posteriores

Desde IBM MQ 8.0, al ejecutarse con la política basada en usuario, algunos mandatos devuelven información diferente respecto a versiones anteriores del producto.

- Los mandatos **dmpmqaut** y **dmpmqcfg** muestran registros basados en usuario, ya que realizan las operaciones equivalentes de PCF.
- El plug-in OAM para IBM MQ Explorer muestra registros basados en usuario y permite modificaciones basadas en usuario.
- La función **Inquire** de OAM devuelve resultados que muestran que tiene capacidad de usuario.

La utilización del atributo **-p** en el mandato **setmqaut** no otorga acceso a todos los usuarios del mismo grupo primario, cuando las autorizaciones basadas en usuario están habilitadas en el archivo `qm.ini` tal como se describe en Stanza de servicio del archivo qm.ini.

Si comienza a utilizar la autorización basada en usuario y tiene muchos usuarios, probablemente, habrá más registros almacenados en la cola AUTH que con el modelo basado en grupo, y el proceso de autorización podría tardar un poco más que antes, ya que hay más registros para verificar. No se espera que este aumento sea significativo. Si es necesario, puede utilizar una combinación de permisos de usuario y grupo.

Consideraciones sobre la migración

Si cambia el modelo de grupo a usuario para un gestor de colas existente, no se produce ningún efecto inmediato. Las autorizaciones que ya se han realizado se siguen aplicando. Cualquier usuario que se conecta al gestor de colas recibe los mismos privilegios que antes: la combinación de todos los grupos a los que pertenece su ID. Cuando se emiten nuevos mandatos **setmqaut** para los ID de usuario, surten efecto de forma inmediata.

Si crea un gestor de colas nuevo con la política de usuario, este gestor de colas solo tiene permisos para el usuario que lo ha creado (que normalmente suele ser, aunque no necesariamente siempre, el ID de usuario `mqm`). También hay permisos que se otorgan automáticamente al grupo `mqm`. Sin embargo, si no tiene `mqm` como grupo principal, el grupo `mqm` no está incluido en el conjunto inicial de autorizaciones.

Si pasa de una política de usuario a una política de grupo, las autorizaciones basadas en usuario no se suprimen automáticamente. Sin embargo, dejan de utilizarse durante la comprobación de permisos. Antes de revertir la política, guarde la configuración actual, cambie la política y reinicie el gestor de colas y, después, reproduzca el script. Puesto que ahora es un gestor de colas basado en grupo, la consecuencia es que las reglas de ID de usuario se almacenan basándose en el grupo principal.

Conceptos relacionados

Gestor de autorizaciones sobre objetos (OAM)

"Principales y grupos en AIX, Linux, and Windows" en la página 413

Los principales pueden pertenecer a grupos. Al otorgar el acceso a recursos a grupos en vez de a usuarios individuales, puede reducir la cantidad de administración necesaria. Las listas de control de acceso (ACL) se basan en grupos y en los ID de usuario.

Referencia relacionada

Stanza de servicio del archivo qm.ini

Mandato **crtmqm** (crear gestor de colas)

ALW Otorgar acceso a un objeto IBM MQ en AIX, Linux, and Windows

Utilice el mandato de control **setmqaut**, el mandato MQSC **SET AUTHREC** o el mandato PCF **MQCMD_SET_AUTH_REC** para otorgar a usuarios, y grupos de usuarios, acceso a los objetos IBM MQ. Tenga en cuenta que en IBM MQ Appliance solo puede utilizar el mandato **SET AUTHREC** .

Para obtener una definición completa del mandato de control **setmqaut** y su sintaxis, consulte setmqaut.

Para obtener una definición completa del mandato MQSC **SET AUTHREC** y su sintaxis, consulte SET AUTHREC.

Para obtener una definición completa del mandato PCF **MQCMD_SET_AUTH_REC** y su sintaxis, consulte Establecer registro de autorización.

El gestor de colas debe estar en ejecución para poder utilizar este mandato. Cuando haya modificado el acceso de un principal, el OAM reflejará inmediatamente los cambios.

Para otorgar a los usuarios acceso a un objeto, debe especificar:

- El nombre del gestor de colas que es el propietario de los objetos con los que está trabajando; si no especifica el nombre de un gestor de colas, se utilizará el gestor de colas predeterminado.
- El nombre y el tipo del objeto (para identificar el objeto de forma exclusiva). El nombre se especifica como un *perfil*; puede ser el nombre explícito del objeto o un nombre genérico que incluya caracteres comodín. Para obtener una descripción detallada de los perfiles genéricos y cómo se utilizan los caracteres comodín en los mismos, consulte el "Utilización de perfiles genéricos del OAM en AIX, Linux, and Windows" en la página 363.
- Uno o varios principales y nombres de grupo a los que se aplica la autorización.

  Si un ID de usuario contiene espacios, póngalo entre signos de interrogación cuando utilice este mandato. En sistemas Windows, puede calificar un ID de usuario con un nombre de dominio. Si el ID de usuario real contiene un símbolo (@), sustitúyalo por @@ para mostrar que forma parte del ID de usuario, no el delimitador entre el ID de usuario y el nombre de dominio.
- Una lista de autorizaciones. Cada elemento de la lista especifica un tipo de acceso que se va a otorgar (o revocar) para este objeto. Cada autorización de la lista se especifica como una palabra clave, con un signo más (+) o un signo menos (-) como prefijo. Utilice un signo más para añadir la autorización especificada y un signo menos para eliminar la autorización. No debe haber ningún espacio entre el signo + o - y la palabra clave.

  Se puede especificar cualquier número de autorizaciones en un solo mandato. Por ejemplo, la lista de autorizaciones que permite que un usuario o un grupo transfiera los mensajes a una cola y los examine pero revoca el acceso para la obtención de mensajes es:

```
+browse -get +put
```

Ejemplos de cómo utilizar el mandato `setmqaut`

Los ejemplos siguientes muestran cómo utilizar el mandato `setmqaut` para otorgar y revocar el permiso para utilizar un objeto:

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

En este ejemplo:

- `saturn.queue.manager` es el nombre del gestor de colas.
- `queue` es el tipo de objeto.
- `RED.LOCAL.QUEUE` es el nombre del objeto.
- `groupa` es el identificador del grupo cuyas autorizaciones se van a modificar.
- `+browse -get +put` es la lista de autorizaciones para la cola especificada.
  - `+browse` añade autorización para examinar los mensajes de la cola (para emitir **MQGET** con la opción browse).
  - `-get` suprime la autorización para obtener (**MQGET**) mensajes de la cola.
  - `+put` añade autorización para transferir (**MQPUT**) mensajes a la cola.

El mandato siguiente revoca la autorización put en la cola MyQueue del principal fvuser y de los grupos groupa y groupb. En sistemas AIX and Linux, este mandato también revoca la autorización de transferencia (put) para todos los principales del mismo grupo primario que fvuser.

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

Utilización del mandato setmqaut con un servicio de autorización diferente

Si utiliza su propio servicio de autorización en lugar del OAM, puede especificar el nombre de este servicio en el mandato **setmqaut** para dirigir el mandato a este servicio. Debe especificar este parámetro si tiene varios componentes instalables que se están ejecutando al mismo tiempo; si no es así, la actualización se realiza en el primer componente instalable del servicio de autorización. De forma predeterminada, es el OAM suministrado.

Notas de uso para SET AUTHREC

La lista de autorizaciones a añadir y la lista de autorizaciones a eliminar no se pueden solapar. Por ejemplo, no puede añadir la autorización de visualización y eliminar la autorización de visualización con el mismo mandato. Esta regla se aplica incluso si las autorizaciones se expresan utilizando opciones distintas. Por ejemplo, el mandato siguiente falla porque la autorización DSP se solapa con la autorización ALLADM:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

La excepción a este comportamiento de solapamiento es con la autorización ALL. El mandato siguiente añade primero las autorizaciones ALL y, a continuación, elimina la autorización SETID:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

El mandato siguiente elimina primero las autorizaciones ALL y, a continuación, añade la autorización DSP:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

Independientemente del orden en el que se proporcionen en el mandato, las ALL se procesan en primer lugar.

ALW Utilización de perfiles genéricos del OAM en AIX, Linux, and Windows

Utilice perfiles genéricos de OAM para establecer, en una sola operación, los privilegios de un usuario para muchos objetos; en lugar de tener que emitir mandatos **setmqaut** o mandatos **SET AUTHREC** separados para cada objeto individual cuando se crea. Tenga en cuenta que en IBM MQ Appliance solo puede utilizar el mandato **SET AUTHREC** .

Si utiliza perfiles genéricos en los mandatos setmqaut oSET AUTHREC podrá establecer una autorización genérica para todos los objetos que se ajusten a este perfil.

Este conjunto de temas describen de forma más detallada el uso de perfiles genéricos.

Utilización de caracteres comodín en perfiles del OAM

Lo que hace que un perfil sea genérico es el uso de caracteres especiales (caracteres comodín) en el nombre del perfil. Por ejemplo, el carácter comodín de signo de interrogación (?) coincide con cualquier carácter individual de un nombre. Por lo tanto, si especifica ABC.?EF, la autorización que concede a este perfil se aplica a cualquier objeto que tenga los nombres ABC.DEF, ABC.CEF, ABC.BEF, etc.

Los caracteres comodín disponibles son:

?

Utilice el signo de interrogación (?) en lugar de cualquier otro carácter. Por ejemplo, `AB.?D` se aplica a los objetos `AB.CD`, `AB.ED` y `AB.FD`.

*****

Utilice el asterisco (*) como:

- Un *calificador* de un nombre de perfil para que coincida con cualquier calificador de un nombre de objeto. Un calificador es la parte de un nombre de objeto delimitada por un punto. Por ejemplo, en `ABC.DEF.GHI`, los calificadores son `ABC`, `DEF` y `GHI`.

  Por ejemplo, `ABC.*.JKL` se aplica a los objetos `ABC.DEF.JKL` y `ABC.GHI.JKL`. (Tenga en cuenta que **no** se aplica a `ABC.JKL`; cuando el asterisco (*) se utiliza en este contexto siempre indica un calificador.)
- Un carácter contenido en un calificador de un nombre de perfil para que coincida con cero o más caracteres incluidos en el calificador de un nombre de objeto.

  Por ejemplo, `ABC.DE*.JKL` se aplica a los objetos `ABC.DE.JKL`, `ABC.DEF.JKL` y `ABC.DEGH.JKL`.

******

Utilice el asterisco doble (**) **una vez** en el nombre de un perfil como:

- El nombre de perfil completo para que coincida con todos los nombres de objetos. Por ejemplo, si utiliza `-t prcs` para identificar procesos y, a continuación, utiliza ** como nombre de perfil, puede cambiar las autorizaciones para todos los procesos.
- Como cualquier calificador del principio, mitad o final de un nombre de perfil para que coincida con cero o más calificadores contenidos en un nombre de objeto. Por ejemplo, `**.ABC` identifica todos los objetos con el calificador final ABC.

Sólo puede utilizar el asterisco doble ** como calificador completo:

```
**.DEF
  ABC.**
  A*.**
```

pero no como

```
A**
```

de lo contrario, recibirá el mensaje AMQ7226E: `El nombre de perfil no es válido.`

Nota: Cuando utilice caracteres comodín en sistemas AIX and Linux, **debe** encerrar el nombre de perfil entre comillas simples.

Prioridades de perfiles

Una cuestión importante que debe comprender cuando utilice perfiles genéricos es la prioridad que se otorga a los perfiles a la hora de decidir qué autorizaciones se han de aplicar a un objeto que se está creando. Por ejemplo, suponga que ha emitido los mandatos:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

El primero otorga autorización de colocación a todas las colas para el principal fred con nombres que coinciden con el perfil AB. *; el segundo otorga autorización de obtención sobre los mismos tipos de cola que coinciden con el perfil AB.C*.

Suponga que ahora crea una cola con el nombre AB.CD. Según las reglas de coincidencia de los caracteres comodín, cualquiera de los mandatos setmqaut se puede aplicar a esta cola. Por lo tanto, la cuestión es ¿tiene autorización para transferir o para obtener?

Para encontrar la respuesta, aplique la regla según la cual cuando varios perfiles pueden aplicarse a un objeto, **sólo se aplica el más específico**. El modo en que se aplica esta regla es comparando los nombres de perfil de izquierda a derecha. Siempre que difieren, un carácter no genérico es más especifico que un

carácter genérico. De este modo, en este ejemplo, la cola AB.CD tiene autorización para **obtener** (AB.C* es más específico que AB.*).

Cuando se comparan caracteres genéricos, el orden de *especificidad* es el siguiente:

1. ?
2. *
3. **

Volcado de valores de perfil

Para obtener una definición completa del mandato de control **dmpmqaut** y su sintaxis, consulte dmpmqaut.

Para obtener una definición completa del mandato MQSC **DISPLAY AUTHREC** y su sintaxis, consulte DISPLAY AUTHREC.

Para obtener una definición completa del mandato de PCF **MQCMD_INQUIRE_AUTH_RECS** y su sintaxis, consulte Consultar registros de autorización.

En los ejemplos siguientes se muestra el uso del mandato de control **dmpmqaut** para volcar registros de autorización para perfiles genéricos:

1. En este ejemplo se vuelcan todos los registros de autorización que tienen un perfil que coincide con la cola a.b.c del principal user1.

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   El volcado resultante es similar al siguiente:

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

   Nota: Aunque los usuarios en AIX and Linux pueden utilizar la opción `-p` para el mandato **dmpmqaut** , deben utilizar `-g groupname` en su lugar al definir autorizaciones.
2. Este ejemplo realiza un volcado de todos los registros de autorización que tienen un perfil que coincide con la cola a.b.c.

   ```
   dmpmqaut -m qmgr1 -n a.b.c -t q
   ```

   El volcado resultante es similar al siguiente:

   ```
   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 ejemplo vuelca todos los registros de autorización para el perfil a.b. *, de tipo cola.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

El volcado resultante es similar al siguiente:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. Este ejemplo realiza un volcado de todos los registros de autorización para el gestor de colas qmX.

```
dmpmqaut -m qmX
```

El volcado resultante es similar al siguiente:

```
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 ejemplo realiza un volcado de todos los nombres de perfil y tipos de objeto para el gestor de colas qmX.

```
dmpmqaut -m qmX -l
```

El volcado resultante es similar al siguiente:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

Nota: Solamente para IBM MQ for Windows, todos los principales visualizados incluyen información de dominio, por ejemplo:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW *Utilización de caracteres comodín en perfiles del OAM en AIX, Linux, and Windows*

Utilice caracteres comodín en un nombre de perfil del gestor de autorizaciones sobre objetos (OAM) para hacer que dicho perfil sea aplicable a más de un objeto.

Lo que hace que un perfil sea genérico es el uso de caracteres especiales (caracteres comodín) en el nombre del perfil. Por ejemplo, el carácter comodín de signo de interrogación (?) coincide con cualquier carácter individual de un nombre. Por lo tanto, si especifica `ABC.?EF`, la autorización que concede a este perfil se aplica a cualquier objeto que tenga los nombres `ABC.DEF`, `ABC.CEF`, `ABC.BEF`, etc.

Los caracteres comodín disponibles son:

?

Utilice el signo de interrogación (?) en lugar de cualquier otro carácter. Por ejemplo, `AB.?D` se aplica a los objetos `AB.CD`, `AB.ED` y `AB.FD`.

*****

Utilice el asterisco (*) como:

- Un *calificador* de un nombre de perfil para que coincida con cualquier calificador de un nombre de objeto. Un calificador es la parte de un nombre de objeto delimitada por un punto. Por ejemplo, en `ABC.DEF.GHI`, los calificadores son `ABC`, `DEF` y `GHI`.

  Por ejemplo, `ABC.*.JKL` se aplica a los objetos `ABC.DEF.JKL` y `ABC.GHI.JKL`. (Tenga en cuenta que **no** se aplica a `ABC.JKL`; cuando el asterisco (*) se utiliza en este contexto siempre indica un calificador.)
- Un carácter contenido en un calificador de un nombre de perfil para que coincida con cero o más caracteres incluidos en el calificador de un nombre de objeto.

  Por ejemplo, `ABC.DE*.JKL` se aplica a los objetos `ABC.DE.JKL`, `ABC.DEF.JKL` y `ABC.DEGH.JKL`.

******

Utilice el asterisco doble (**) **una vez** en el nombre de un perfil como:

- El nombre de perfil completo para que coincida con todos los nombres de objetos. Por ejemplo, si utiliza `-t prcs` para identificar procesos y, a continuación, utiliza ** como nombre de perfil, puede cambiar las autorizaciones para todos los procesos.
- Como cualquier calificador del principio, mitad o final de un nombre de perfil para que coincida con cero o más calificadores contenidos en un nombre de objeto. Por ejemplo, `**.ABC` identifica todos los objetos con el calificador final ABC.

Nota: Cuando utilice caracteres comodín en sistemas AIX and Linux, **debe** encerrar el nombre de perfil entre comillas simples.

ALW *Prioridades de perfiles en AIX, Linux, and Windows*

Se puede aplicar más de un perfil genérico a un único objeto. Cuando este sea el caso, se aplica la regla más específica.

Una cuestión importante que debe comprender cuando utilice perfiles genéricos es la prioridad que se otorga a los perfiles a la hora de decidir qué autorizaciones se han de aplicar a un objeto que se está creando. Por ejemplo, suponga que ha emitido los mandatos:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

El primero otorga autorización de colocación a todas las colas para el principal fred con nombres que coinciden con el perfil AB. *; el segundo otorga autorización de obtención sobre los mismos tipos de cola que coinciden con el perfil AB.C*.

Suponga que ahora crea una cola con el nombre AB.CD. Según las reglas de coincidencia de los caracteres comodín, cualquiera de los mandatos setmqaut se puede aplicar a esta cola. Por lo tanto, la cuestión es ¿tiene autorización para transferir o para obtener?

Para encontrar la respuesta, aplique la regla según la cual cuando varios perfiles pueden aplicarse a un objeto, **sólo se aplica el más específico**. El modo en que se aplica esta regla es comparando los nombres de perfil de izquierda a derecha. Siempre que difieren, un carácter no genérico es más especifico que un carácter genérico. De este modo, en este ejemplo, la cola AB.CD tiene autorización para **obtener** (AB.C* es más específico que AB.*).

Cuando se comparan caracteres genéricos, el orden de *especificidad* es el siguiente:

1. ?
2. *
3. **

Consulte SET AUTHREC para obtener la información equivalente cuando se utiliza este mandatos MQSC.

ALW *Volcado de valores de perfil en AIX, Linux, and Windows*

Utilice el mandato de control **dmpmqaut** , el mandato MQSC **DISPLAY AUTHREC** o el mandato PCF **MQCMD_INQUIRE_AUTH_RECS** para volcar las autorizaciones actuales asociadas con un perfil especificado. Tenga en cuenta que en IBM MQ Appliance solo puede utilizar el mandato **DISPLAY AUTHREC** .

Para obtener una definición completa del mandato de control **dmpmqaut** y su sintaxis, consulte dmpmqaut.

Para obtener una definición completa del mandato MQSC **DISPLAY AUTHREC** y su sintaxis, consulte DISPLAY AUTHREC.

Para obtener una definición completa del mandato de PCF **MQCMD_INQUIRE_AUTH_RECS** y su sintaxis, consulte Consultar registros de autorización.

En los ejemplos siguientes se muestra el uso del mandato de control **dmpmqaut** para volcar registros de autorización para perfiles genéricos:

1. En este ejemplo se vuelcan todos los registros de autorización que tienen un perfil que coincide con la cola a.b.c del principal user1.

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   El volcado resultante es similar al ejemplo siguiente:

   ```
   profile:     a.b.*
   object type: queue
   entity:      user1
   type:        principal
   authority:   get, browse, put, inq
   ```

   Nota: Los usuarios de AIX and Linux no pueden utilizar la opción `-p` ; en su lugar, deben utilizar `-g groupname` .
2. Este ejemplo realiza un volcado de todos los registros de autorización que tienen un perfil que coincide con la cola a.b.c.

   ```
   dmpmqaut -m qmgr1 -n a.b.c -t q
   ```

   El volcado resultante es similar al ejemplo siguiente:

   ```
   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 ejemplo vuelca todos los registros de autorización para el perfil a.b. *, de tipo cola.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

   El volcado resultante es similar al ejemplo siguiente:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. Este ejemplo realiza un volcado de todos los registros de autorización para el gestor de colas qmX.

```
dmpmqaut -m qmX
```

   El volcado resultante es similar al ejemplo siguiente:

```
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 ejemplo realiza un volcado de todos los nombres de perfil y tipos de objeto para el gestor de colas qmX.

```
dmpmqaut -m qmX -l
```

   El volcado resultante es similar al ejemplo siguiente:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

Nota: Solamente para IBM MQ for Windows, todos los principales visualizados incluyen información de dominio, por ejemplo:

```
profile:     a.b.*
```

```
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW Visualización de los valores de acceso en AIX, Linux, and Windows

Utilice el mandato de control **dspmqaut** , el mandato MQSC **DISPLAY AUTHREC** o el mandato PCF **MQCMD_INQUIRE_ENTITY_AUTH** para ver las autorizaciones que tiene un principal o grupo específico para un objeto determinado. Tenga en cuenta que, en IBM MQ Appliance, solamente puede utilizar el mandato **DISPLAY AUTHREC**.

El gestor de colas debe estar en ejecución para poder utilizar este mandato. Cuando modifique el acceso de un principal, el OAM reflejará inmediatamente los cambios. Las autorizaciones sólo pueden visualizarse para un grupo o principal cada vez.

Para obtener una definición completa del mandato de control **dmpmqaut** y su sintaxis, consulte dmpmqaut.

Para obtener una definición completa del mandato MQSC **DISPLAY AUTHREC** y su sintaxis, consulte DISPLAY AUTHREC.

Para obtener una definición completa del mandato de PCF **MQCMD_INQUIRE_AUTH_RECS** y su sintaxis, consulte Consultar registros de autorización.

El ejemplo siguiente muestra el uso del mandato de control **dspmqaut** para visualizar las autorizaciones que tiene el grupo `GpAdmin` para una definición de proceso llamada `Annuities` que está en gestor de colas `QueueMan1`.

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

ALW Modificación y revocación del acceso a un objeto de IBM MQ en AIX, Linux, and Windows

Para modificar el nivel de acceso que un usuario o grupo tiene para un objeto, utilice el mandato de control **setmqaut**, el mandato MQSC **DELETE AUTHREC** del mandato PCF**MQCMD_DELETE_AUTH_REC**.

MQ Appliance Tenga en cuenta que en IBM MQ Appliance, solo puede utilizar el mandato **DELETE AUTHREC**.

El proceso de eliminación de un usuario de un grupo se describe en:

- Windows "Creación y gestión de grupos en Windows" en la página 154
- AIX "Creación y gestión de grupos en AIX" en la página 153
- Linux "Creación y gestión de grupos en Linux" en la página 154

.

Al ID de usuario que crea un objeto IBM MQ se le otorga autorizaciones de control totales para dicho objeto. Si elimina este ID de usuario del grupo mqm local (o del grupo Administradores en sistemas Windows), no se revocarán estas autorizaciones. Utilice el mandato de control **setmqaut** o el mandato PCF **MQCMD_DELETE_AUTH_REC** para revocar el acceso a un objeto para el ID de usuario que lo ha creado, después de eliminarlo del grupo mqm o del grupo Administradores.

Para obtener una definición completa del mandato de control setmqaut y su sintaxis, consulte setmqaut.

Para obtener una definición completa del mandato MQSC **DELETE AUTHREC** y su sintaxis, consulte DELETE AUTHREC.

Para obtener una definición completa del mandato PCF **MQCMD_DELETE_AUTH_REC** y su sintaxis, consulte Suprimir registro de autorización.

Windows En Windows, a partir de IBM MQ 8.0, puede suprimir las entradas OAM correspondientes a una cuenta de usuario de Windows concreta en cualquier momento utilizando el parámetro **-u** *SID* de **setmqaut**.

Antes de IBM MQ 8.0, tenía que suprimir las entradas OAM correspondientes a una cuenta de usuario de Windows concreta antes de suprimir el perfil de usuario. No se podían eliminar las entradas OAM después de eliminar la cuenta de usuario.

ALW Impedir comprobaciones de acceso de seguridad en los sistemas AIX, Linux, and Windows

Nota: en este tema se describe la funcionalidad que no se recomienda habilitar. Para desactivar la comprobación de seguridad, puede inhabilitar el gestor de autorizaciones sobre objetos (OAM). Esta acción podría resultar adecuada en un entorno de prueba. Cuando está inhabilitado, el gestor de colas ya no puede realizar comprobaciones de autenticación de autorización o conexión. Se pueden seguir utilizando TLS, los registros de autenticación de canal y las salidas de seguridad. Al haber inhabilitado o eliminado el OAM, no puede añadir un OAM a un gestor de colas existente.

Si decide que no desea realizar comprobaciones de seguridad (por ejemplo, en un entorno de prueba), puede inhabilitar el OAM de dos modos:

- Antes de crear un gestor de colas, establezca la variable de entorno del sistema operativo **MQSNOAUT**.

  Para obtener información sobre las implicaciones de establecer la variable de entorno **MQSNOAUT** y cómo establecer **MQSNOAUT** en AIX, Linux, and Windows, consulte Descripciones de variables de entorno.
- Edite el archivo de configuración del gestor de colas para eliminar el servicio.

Aviso: Cuando se elimina un gestor de autorizaciones sobre objetos, no se puede volver a colocar en un gestor de colas existente. Esto se debe a que el OAM debe estar en su sitio cuando se crea el objeto. Para utilizar el OAM de IBM MQ después de haberlo eliminado, reconstruir el gestor de colas.

Si utiliza **setmqaut** o el mandato **dspmqaut** mientras el OAM está inhabilitado, tenga en cuenta los puntos siguientes:

- El OAM no validará el principal, o grupo, especificado lo que significa que el mandato puede aceptar valores no válidos.
- El OAM no realiza comprobaciones de seguridad e indica que todos los principales y grupos están autorizado a realizar todas las operaciones aplicables de objetos.
- Las credenciales pasadas al OAM para comprobaciones de autenticación no se validan.

Conceptos relacionados

Servicios y componentes instalables para AIX, Linux, and Windows

Tareas relacionadas

Configuración de servicios instalables

Referencia relacionada

Información de referencia de servicios instalables

Otorgar el acceso necesario a los recursos

Utilice este tema para determinar qué tareas deben realizarse para aplicar la seguridad a su sistema IBM MQ.

Acerca de esta tarea

Durante esta tarea se decide qué acciones son necesarias para aplicar el nivel de seguridad apropiado a los elementos de su instalación IBM MQ. Cada tarea a la que se refiere ofrece instrucciones paso a paso para todas las plataformas.

Procedimiento

1. ¿Necesita limitar el acceso a su gestor de colas a determinados usuarios?
   a) No: No realice ninguna acción más.
   b) Sí: Vaya hasta la siguiente pregunta.
2. ¿Estos usuarios necesitan acceso de administrador parcial sobre un subconjunto de recursos del gestor de colas?
   a) No: Vaya hasta la siguiente pregunta.
   b) Sí: Consulte "Cómo otorgar acceso de administrador parcial sobre un subconjunto de recursos del gestor de colas" en la página 372.
3. ¿Estos usuarios necesitan acceso de administrador total sobre un subconjunto de recursos de gestor de colas?
   a) No: Vaya hasta la siguiente pregunta.
   b) Sí: Consulte "Cómo otorgar acceso de administrador total sobre un subconjunto de recursos del gestor de colas" en la página 381.
4. ¿Estos usuarios necesitan acceso de sólo lectura a todos los recursos del gestor de colas?
   a) No: Vaya hasta la siguiente pregunta.
   b) Sí: Consulte "Otorgar acceso de sólo lectura a todos los recursos de un gestor de colas" en la página 387.
5. ¿Estos usuarios necesitan acceso de administrador total sobre todos los recursos del gestor de colas?
   a) No: Vaya hasta la siguiente pregunta.
   b) Sí: Consulte "Otorgar acceso administrativo completo a todos los recursos de un gestor de colas" en la página 388.
6. ¿Necesita que las aplicaciones de usuario se conecten con su gestor de colas?
   a) No: Inhabilite la conectividad, tal como se describe en "Eliminar la conectividad con el gestor de colas" en la página 389
   b) Sí: Consulte "Cómo permitir que las aplicaciones de usuario se conecten con su gestor de colas" en la página 390.

Multi z/OS Cómo otorgar acceso de administrador parcial sobre un subconjunto de recursos del gestor de colas

Necesita otorgar a algunos usuarios acceso parcial de administrador a algunos, pero no todos, de los recursos del gestor de colas. Utilice esta tabla para determinar las acciones que necesita llevar a cabo.

Tabla 72. Cómo otorgar acceso de administrador parcial a un subconjunto de recursos del gestor de colas

Los usuarios necesitan administrar objetos de este tipo	**Realice esta acción**
Colas	Otorgue acceso de administrador parcial a las colas necesarias, tal como se describe en "Otorgar acceso administrativo limitado a algunas colas" en la página 373
Temas	Otorgue acceso de administrador parcial a los temas necesarios, tal como se describe en "Otorgar acceso administrativo limitado a algunos temas" en la página 374
Canales	Otorgue acceso de administrador parcial a los canales necesarios, tal como se describe en "Otorgar acceso administrativo limitado a algunos canales" en la página 375

Tabla 72. Cómo otorgar acceso de administrador parcial a un subconjunto de recursos del gestor de colas (continuación)

Los usuarios necesitan administrar objetos de este tipo	**Realice esta acción**
El gestor de colas	Otorgue acceso de administrador parcial al gestor de colas, tal como se describe en "Otorgar acceso administrativo parcial a un gestor de colas" en la página 376
todos los Procesos	Otorgue acceso de administrador parcial a los procesos necesarios, tal como se describe en "Otorgar acceso administrativo limitado a algunos procesos" en la página 378
Listas de nombres	Otorgue acceso de administrador parcial a las listas de nombres necesarias, tal como se describe en "Otorgar acceso administrativo limitado a algunas listas de nombres" en la página 379
Servicios	Otorgue acceso de administrador parcial a los servicios necesarios, tal como se describe en "Otorgar acceso administrativo limitado a algunos servicios" en la página 380

Otorgar acceso administrativo limitado a algunas colas

Otorgue acceso administrativo parcial a algunas colas en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo limitado a algunas colas, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Nota: MQ Appliance En IBM MQ Appliance puede utilizar solo el mandato **SET AUTHREC**.

Procedimiento

- ALW
  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
  ```

- IBM i
  Para IBM i, emita el mandato siguiente:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS Para z/OS, emita los mandatos siguientes para otorgar acceso a una cola especificada:

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

Para especificar qué mandatos MQSC puede ejecutar el usuario en la cola, emita los mandatos siguientes para cada mandato MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction. QType UACC(NONE)
PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Para permitir al usuario utilizar el mandato DISPLAY QUEUE, emita los mandatos siguientes:

```
RDEFINE MQCMDS QMgrName.DISPLAY. QType UACC(NONE)
PERMIT QMgrName.DISPLAY. QType CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas.

z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

ObjectProfile
El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

ReqdAction
La acción que va a permitir realizar al grupo:

- ALW En los sistemas AIX, Linux, and Windows, cualquier combinación de las autorizaciones siguientes: +chg, +clr, +dlt, +dsp. La autorización +alladm es equivalente a +chg +clr +dlt +dsp.
- IBM i En sistemas IBM i, cualquier combinación de las siguientes autorizaciones: *ADMCHG, *ADMCLR, *ADMDLT, *ADMDSP. La autorización *ALLADM es equivalente a todas estas autorizaciones individuales.
- z/OS En z/OS, uno de los valores ALTER, CLEAR, DELETE, o MOVE.

Nota: Otorgar +crt para las colas convierte indirectamente al usuario o grupo en un administrador. No utilice la autorización +crt para otorgar acceso administrativo limitado a algunas colas.

QType

Para el mandato DISPLAY, uno de los valores QUEUE, QLOCAL, QALIAS, QMODEL, QREMOTE o QCLUSTER.

Para otros valores de *AcciónReq*, uno de los valores QLOCAL, QALIAS, QMODEL o QREMOTE.

Otorgar acceso administrativo limitado a algunos temas

Otorgue acceso administrativo parcial a algunos temas en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo limitado a algunos temas, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
```

- IBM i
  Para IBM i, emita el mandato siguiente:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS Para z/OS, emita los mandatos siguientes:

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Estos mandatos otorgan acceso al tema especificado. Para determinar qué mandatos MQSC puede realizar el usuario en el tema, emita los mandatos siguientes para cada mandato MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.TOPIC UACC(NONE)
PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Para permitir al usuario utilizar el mandato DISPLAY TOPIC, emita los mandatos siguientes:

```
RDEFINE MQCMDS QMgrName.DISPLAY.TOPIC UACC(NONE)
PERMIT QMgrName.DISPLAY.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

  ReqdAction
  La acción que va a permitir realizar al grupo:

  – ALW En sistemas AIX, Linux, and Windows , cualquier combinación de las autorizaciones siguientes: + chg, + clr, + crt, + dlt, + dsp. + ctrl. La autorización +alladm es equivalente a +chg +clr +dlt +dsp.

  – IBM i En IBM i, cualquier combinación de las siguientes autorizaciones: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL. La autorización *ALLADM es equivalente a todas estas autorizaciones individuales.

  – z/OS En z/OS, uno de los valores ALTER, CLEAR, DEFINE, DELETE o MOVE.

Otorgar acceso administrativo limitado a algunos canales

Otorgue acceso administrativo parcial a algunos canales en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo limitado a algunos canales, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  En AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
```

- IBM i
  En IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS En z/OS:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Estos mandatos otorgan acceso al canal especificado. Para determinar qué mandatos MQSC puede realizar el usuario en el canal, emita los mandatos siguientes para cada mandato MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.CHANNEL UACC(NONE)
PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Para permitir al usuario utilizar el mandato DISPLAY CHANNEL, emita los mandatos siguientes:

```
RDEFINE MQCMDS QMgrName.DISPLAY.CHANNEL UACC(NONE)
PERMIT QMgrName.DISPLAY.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

  ReqdAction
  La acción que va a permitir realizar al grupo:

  - ALW En AIX, Linux, and Windows, cualquier combinación de las autorizaciones siguientes: + chg, + clr, + crt, + dlt, + dsp. + ctrl, + ctrlx. La autorización +alladm es equivalente a +chg +clr +dlt +dsp.
  - IBM i En IBM i, cualquier combinación de las siguientes autorizaciones: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. La autorización *ALLADM es equivalente a todas estas autorizaciones individuales.
  - z/OS En z/OS, uno de los valores ALTER, CLEAR, DEFINE, DELETE o MOVE.

Otorgar acceso administrativo parcial a un gestor de colas

Otorgue acceso administrativo parcial a un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo limitado para llevar a cabo algunas acciones en el gestor de colas, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  En AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
  ```

- IBM i

  En IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS

  En z/OS:

  Para determinar qué mandatos MQSC puede realizar en el gestor de colas, emita los mandatos siguientes para cada mandato MQSC:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.QMGR UACC(NONE)
  PERMIT QMgrName. ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Para permitir al usuario utilizar el mandato DISPLAY QMGR, emita los mandatos siguientes:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
  PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

  ReqdAction
  La acción que va a permitir realizar al grupo:

  – ALW En AIX, Linux, and Windows, cualquier combinación de las siguientes autorizaciones: +chg, +clr, +crt, +dlt, +dsp. La autorización +alladm es equivalente a +chg +clr +dlt +dsp.

    Aunque +set es una autorización de MQI y no normalmente no se considera administrativa, otorgar +set en el gestor de colas puede conducir indirectamente a la autorización administrativa completa. No otorgue +set a usuarios y aplicaciones ordinarios.

  – IBM i En sistemas IBM i, cualquier combinación de las siguientes autorizaciones: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. La autorización *ALLADM es equivalente a todas estas autorizaciones individuales.

Otorgar acceso administrativo limitado a algunos procesos

Otorgue acceso administrativo parcial a algunos procesos en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo limitado a algunos procesos, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  En AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
  ```

- IBM i

  En IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS En z/OS:

  ```
  RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
  PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Estos mandatos otorgan acceso al canal especificado. Para determinar qué mandatos MQSC puede realizar el usuario en el canal, emita los mandatos siguientes para cada mandato MQSC:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.PROCESS UACC(NONE)
  PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Para permitir al usuario utilizar el mandato DISPLAY PROCESS, emita los mandatos siguientes:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.PROCESS UACC(NONE)
  PERMIT QMgrName.DISPLAY.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

  ReqdAction
  La acción que va a permitir realizar al grupo:

  – ALW En AIX, Linux, and Windows, cualquier combinación de las siguientes autorizaciones: +chg, +clr, +crt, +dlt, +dsp. La autorización +alladm es equivalente a +chg +clr +dlt +dsp.

– IBM i En sistemas IBM i, cualquier combinación de las siguientes autorizaciones: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. La autorización *ALLADM es equivalente a todas estas autorizaciones individuales.

– z/OS En z/OS, uno de los valores ALTER, CLEAR, DEFINE, DELETE o MOVE.

Otorgar acceso administrativo limitado a algunas listas de nombres

Otorgue acceso administrativo parcial a algunas listas de nombres en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo limitado a algunas listas de nombres, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  En AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
```

- IBM i
  En IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS En z/OS:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Estos mandatos otorgan acceso a la lista de nombres especificada. Para determinar qué mandatos MQSC puede realizar el usuario en la lista de nombres, emita los mandatos siguientes para cada mandato MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.NAMELIST UACC(NONE)
PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Para permitir al usuario utilizar el mandato DISPLAY NAMELIST, emita los mandatos siguientes:

```
RDEFINE MQCMDS QMgrName.DISPLAY.NAMELIST UACC(NONE)
PERMIT QMgrName.DISPLAY.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

ReqdAction
La acción que va a permitir realizar al grupo:

- ALW En AIX, Linux, and Windows, cualquier combinación de las siguientes autorizaciones: +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp. La autorización +alladm es equivalente a +chg +clr +dlt +dsp.
- IBM i En IBM i, cualquier combinación de las siguientes autorizaciones: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. La autorización *ALLADM es equivalente a todas estas autorizaciones individuales.
- z/OS En z/OS, uno de los valores ALTER, CLEAR, DEFINE, DELETE o MOVE.

Otorgar acceso administrativo limitado a algunos servicios

Otorgue acceso administrativo parcial a algunos servicios en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso de administrador limitado a algunas acciones, utilice los mandatos apropiados para su sistema operativo. z/OS Tenga en cuenta que los objetos de servicio no existen en z/OS.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  En AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- En IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS En z/OS:

  Estos mandatos otorgan acceso al servicio especificado. Para determinar qué mandatos MQSC puede realizar el usuario en el servicio, emita los mandatos siguientes para cada mandato MQSC:

```
RDEFINE MQCMDS QMgrName. ReqdAction.SERVICE UACC(NONE)
PERMIT QMgrName. ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Para permitir al usuario utilizar el mandato DISPLAY SERVICE, emita los mandatos siguientes:

```
RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

ReqdAction
La acción que va a permitir realizar al grupo:

- ALW En los sistemas AIX, Linux, and Windows, cualquier combinación de las autorizaciones siguientes: +chg, +clr, +crt, +dlt, +ctrl, +ctrlx, +dsp. La autorización +alladm es equivalente a +chg +clr +dlt +dsp.
- IBM i En IBM i, cualquier combinación de las siguientes autorizaciones: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. La autorización *ALLADM es equivalente a todas estas autorizaciones individuales.

Cómo otorgar acceso de administrador total sobre un subconjunto de recursos del gestor de colas

Necesita otorgar a algunos usuarios acceso completo de administrador a algunos, pero no todos, de los recursos del gestor de colas. Utilice estas tablas para determinar las acciones que necesita llevar a cabo.

Tabla 73. Cómo otorgar acceso de administrador total a un subconjunto de recursos del gestor de colas

Los usuarios necesitan administrar objetos de este tipo	**Realice esta acción**
Colas	Otorgue acceso de administrador total a las colas necesarias, tal como se describe en “Otorgar acceso administrativo completo a algunas colas” en la página 381
Temas	Otorgue acceso de administrador total a los temas necesarios, tal como se describe en “Otorgar acceso administrativo completo a algunos temas” en la página 382
Canales	Otorgue acceso de administrador total a los canales necesarios, tal como se describe en “Otorgar acceso administrativo completo a algunos canales” en la página 383
El gestor de colas	Otorgue acceso de administrador total al gestor de colas, tal como se describe en “Otorgar acceso administrativo completo a un gestor de colas” en la página 384
todos los Procesos	Otorgue acceso de administrador total a los procesos necesarios, tal como se describe en “Otorgar acceso administrativo completo a algunos procesos” en la página 384
Listas de nombres	Otorgue acceso de administrador total a las listas de nombres necesarias, tal como se describe en “Otorgar acceso administrativo completo a algunas listas de nombres” en la página 385
Servicios	Otorgue acceso de administrador total a los servicios necesarios, tal como se describe en “Otorgar acceso administrativo completo a algunos servicios” en la página 386

Otorgar acceso administrativo completo a algunas colas

Otorgue acceso administrativo completo a algunas colas en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo completo a algunas colas, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  En AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
  ```

- IBM i

  En IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrName
  ')
  ```

- z/OS

  En z/OS:

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar acceso administrativo completo a algunos temas

Otorgue acceso administrativo completo a algunos temas en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo completo a algunos temas, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  En AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
  ```

- IBM i

  En IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('
QMgrName ')
```

- z/OS

  En z/OS:

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas.

z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

ObjectProfile
El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

Otorgar acceso administrativo completo a algunos canales

Otorgue acceso administrativo completo a algunos canales en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo completo a algunos canales, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  En AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
```

- IBM i

  En IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('
QMgrName ')
```

- z/OS

  En z/OS:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas.

z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

ObjectProfile
El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

Otorgar acceso administrativo completo a un gestor de colas

Otorgue acceso administrativo completo a un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo completo al gestor de colas, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  En AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +alladm
  ```

- IBM i
  En IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  En z/OS:

  ```
  RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
  PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar acceso administrativo completo a algunos procesos

Otorgue acceso administrativo completo a algunos procesos en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo completo a algunos procesos, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  En AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
```

- IBM i
  En IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  En z/OS:

```
RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar acceso administrativo completo a algunas listas de nombres

Otorgue acceso administrativo completo a algunas listas de nombres en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo completo a algunas listas de nombres, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  En AIX, Linux, and Windows:

```
setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
```

- IBM i
  En IBM i:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  En z/OS:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas.

z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

ObjectProfile
El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

Otorgar acceso administrativo completo a algunos servicios

Otorgue acceso administrativo completo a algunos servicios en un gestor de colas, a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar acceso administrativo completo a algunos servicios, utilice los mandatos apropiados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  En AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
  ```

- IBM i
  En IBM i:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  En z/OS:

  ```
  RDEFINE MQADMIN QMgrName.SERVICE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.SERVICE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar acceso de sólo lectura a todos los recursos de un gestor de colas

Otorgue acceso de sólo lectura a todos los recursos de un gestor de colas para cada usuario o grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Utilice el asistente para añadir autorizaciones basadas en funciones o los mandatos correspondientes para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Después de haber cambiado los detalles de autorización, realice una renovación de seguridad utilizando el mandato REFRESH SECURITY .

Procedimiento

- Utilización del asistente:
  a) En el panel del navegador de IBM MQ Explorer, pulse con el botón derecho del ratón en el gestor de colas y pulse **Autorizaciones de objetos** > **Añadir autorizaciones basadas en funciones**
  Se abre el asistente Añadir autorizaciones basadas en funciones.
- ALW
  Para sistemas AIX, Linux, and Windows, emita los mandatos siguientes:

```
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
```

  Las autorizaciones específicas para SYSTEM.ADMIN.COMMAND.QUEUE y SYSTEM.MQEXPLORER.REPLY.MODEL sólo es necesario si desea utilizar IBM MQ Explorer.
- IBM i
  Para IBM i, emita los mandatos siguientes:

```
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 los mandatos siguientes:

```
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)
```

Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas.

z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

Otorgar acceso administrativo completo a todos los recursos de un gestor de colas

Otorgue acceso administrativo completo a todos los recursos de un gestor de colas para cada usuario o grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Puede utilizar el asistente Añadir autorizaciones basadas en roles o los mandatos adecuados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Notas: ALW

1. Si utiliza **runmqsc** para administrar el gestor de colas en lugar de IBM MQ Explorer, debe otorgar autorización para consultar, obtener y examinar SYSTEM.MQSC.REPLY.QUEUE, y no es necesario que otorgue ninguna autorización sobre SYSTEM.MQEXPLORER.REPLY.MODEL .
2. Al otorgar a un usuario acceso a todos los recursos de un gestor de colas, hay algunos mandatos que el usuario no puede ejecutar, a menos que dicho usuario tenga acceso de lectura al archivo `qm.ini` . Esto se debe a las restricciones sobre los usuarios que no son de `mqm` que pueden leer el archivo `qm.ini` .

   El usuario no puede emitir los mandatos siguientes a menos que haya otorgado a dicho usuario acceso de lectura al archivo qm.ini :

   - Definición de un canal que está configurado para utilizar TLS
   - Definición de un canal utilizando variables de inserción de configuración automática definidas en `qm.ini`

Procedimiento

- Si está utilizando el asistente, en el panel IBM MQ Explorer Navigator , pulse con el botón derecho del ratón en el gestor de colas y pulse **Autorizaciones de objeto** > **Añadir autorizaciones basadas en roles**.

  Se abre el asistente Añadir autorizaciones basadas en funciones.
- Linux AIX

  Para sistemas AIX and Linux, emita los mandatos siguientes:

```
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 obtener más información sobre `@class`

- Windows

  Para sistemas Windows, emita los mismos mandatos que para los sistemas AIX and Linux pero utilizando el nombre de perfil @CLASS en lugar de @class.

- IBM i

  Para IBM i, emita el mandato siguiente:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER(' GroupName ') AUT(*ALLADM) MQMNAME(' QMgrName ')
```

- z/OS

  Para z/OS, emita los mandatos siguientes:

```
RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas.

z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

Eliminar la conectividad con el gestor de colas

Si no desea que las aplicaciones de usuario se conecten con el gestor de colas, elimine su autorización para conectarse a él.

Acerca de esta tarea

Revoque la autorización de todos los usuarios a conectarse con el gestor de colas mediante el mandato adecuado para su sistema operativo.

En Multiplatforms, también puede utilizar el mandato DELETE AUTHREC .

Nota: En IBM MQ Appliance solamente puede utilizar el mandato **DELETE AUTHREC**.

Procedimiento

- ALW

  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

```
setmqaut -m QMgrName -t qmgr -g GroupName -connect
```

- IBM i

  Para IBM i, emita el mandato siguiente:

  ```
  RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
  ```

- z/OS

  Para z/OS, emita los mandatos siguientes:

  ```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
  RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
  RDEFINE MQCONN QMgrName.CICS UACC(NONE)
  RDEFINE MQCONN QMgrName.IMS UACC(NONE)
  ```

  No emita ningún mandato PERMIT.

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas.

  z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  GroupName
  Nombre del grupo al que se va a negar el acceso.

Cómo permitir que las aplicaciones de usuario se conecten con su gestor de colas

Desea permitir que la aplicación de usuario se conecte con su gestor de colas. Utilice las tablas de este tema para determinar qué acciones deben llevarse a cabo.

En primer lugar determine si las aplicaciones de cliente se conectarán con su gestor de colas.

Si ninguna de las aplicaciones que se conectarán a su gestor de colas es una aplicación de cliente, inhabilite el acceso remoto tal como se describe en “Inhabilitar el acceso remoto al gestor de colas” en la página 398.

Si una o más de las aplicaciones que se conectarán a su gestor de colas son aplicaciones de cliente, asegure la conectividad remota tal como se describe en “Cómo proteger la conectividad remota con el gestor de colas” en la página 391.

En ambos casos, establezca la seguridad de la conexión tal como se describe en “Configurar la seguridad de conexión” en la página 398

Si desea controlar el acceso a los recursos para cada usuario que se conecta con el gestor de colas, consulte la tabla siguiente. Si la declaración de la primera columna es true, lleve a cabo la acción que aparece en la segunda columna.

Sentencia	Realice esta acción
Tiene aplicaciones que utilizan colas	Consulte “Control del acceso de los usuarios a las colas” en la página 399.
Tiene aplicaciones que utilizan temas	Consulte “Control del acceso de los usuarios a los temas” en la página 405.
Tiene aplicaciones que consultan en el objeto del gestor de colas	Consulte “Otorgar autorización para consultar en un gestor de colas” en la página 406.
Tiene aplicaciones que utilizan objetos de procesos	Consulte “Otorgar autorización para acceder a procesos” en la página 407.
Tiene aplicaciones que utilizan listas de nombres	Consulte “Otorgar autorización para acceder a listas de nombres” en la página 408.

Cómo proteger la conectividad remota con el gestor de colas

Puede proteger la conectividad remota con el gestor de colas utilizando TLS, una salida de seguridad, registros de autenticación de canal o una combinación de estos métodos.

Acerca de esta tarea

Puede conectar un cliente con el gestor de colas utilizando un canal de conexión de cliente en la estación de trabajo cliente y un canal de conexión de servidor en el servidor. Proteja estas conexiones de una de las siguientes maneras.

Procedimiento

1. Utilizando TLS con registros de autenticación de canal:
   a) Impida que cualquier Nombre distinguido (DN) abra un canal, utilizando un registro de autenticación de canal SSLPEERMAP para correlacionar todos los DN con USERSRC(NOACCESS).
   b) Permita que Nombres distinguidos (DN) o conjuntos de DNs específicos abran un canal, utilizando un registro de autenticación de canal SSLPEERMAP para correlacionarlos con USERSRC(CANAL).
2. Utilizando TLS con una salida de seguridad:
   a) Establezca MCAUSER en el canal de conexión de servidor en un identificador de usuario sin privilegios.
   b) Escriba una salida de seguridad para asignar un valor MCAUSER en función del valor del DN TLS que reciba en los campos SSLPeerNamePtr y SSLPeerNameLength que se pasan a la salida en la estructura MQCD.
3. Utilizando TLS con valores de definición de canal fijos:
   a) Establezca SSLPEER en el canal de conexión de servidor en un valor o un rango reducido de valores específico.
   b) Establezca MCAUSER en el canal de conexión de servidor en el ID de usuario con el que debe ejecutarse el canal.
4. Utilizando registros de autenticación de canal en canales que no utilizan TLS:
   a) Impida que cualquier dirección IP abra canales, utilizando un registro de autenticación de canal de correlación de direcciones con ADDRESS(*) y USERSRC(NOACCESS).
   b) Permita que direcciones IP específicas abran canales, utilizando registros de autenticación de canal de correlación de direcciones para esas direcciones con USERSRC(CHANNEL).
5. Utilizando una salida de seguridad:
   a) Escriba una salida de seguridad para autorizar conexiones basadas en la propiedad que elija, por ejemplo la dirección IP de origen.
6. También es posible utilizar registros de autenticación de canal con una salida de seguridad, o utilizar los tres métodos, si sus circunstancias específicas lo exigen.

Bloquear direcciones IP específicas

Puede impedir que un canal específico acepte una conexión entrante de una dirección IP o impedir que el gestor de colas en su conjunto permita el acceso desde una dirección IP, utilizando un registro de autenticación de canal.

Antes de empezar

Habilite los registros de autenticación de canal ejecutando el mandato siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Acerca de esta tarea

Para no permitir que canales específicos acepten una conexión de entrada y garantizar que las conexiones sólo se acepten cuando se utilice el nombre de canal correcto, se puede utilizar un tipo

de regla para bloquear direcciones IP. Para no permitir que una dirección IP acceda al gestor de colas en su conjunto, lo haría normalmente utilizando un cortafuegos para bloquearla permanentemente. No obstante, se puede utilizar otro tipo de regla para permitirle bloquear unas pocas direcciones temporalmente, por ejemplo mientras espera a que se actualice el cortafuegos.

Procedimiento

- Para impedir que las direcciones IP utilicen un canal específico, establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record**.

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

Este mandato tiene tres partes:

SET CHLAUTH (*nombre-canal-genérico*)
Esta parte del mandato se utiliza para controlar si desea bloquear una conexión para todo el gestor de colas, un único canal o un rango de canales. Lo que se especifica aquí determina qué áreas se cubren.

Por ejemplo:

- SET CHLAUTH('*') - bloquea todos los canales de un gestor de colas, es decir, todo el gestor de colas
- SET CHLAUTH('SYSTEM.*') - bloque todos los canales que empiezan por SYSTEM.
- SET CHLAUTH('SYSTEM.DEF.SVRCONN') - bloque el canal SYSTEM.DEF.SVRCONN

Tipo de regla CHLAUTH
Utilice esta parte del mandato para especificar el tipo de mandato y determinar si desea proporcionar una sola dirección o una lista de direcciones.

Por ejemplo:

- TYPE(ADDRESSMAP) - Use ADDRESSMAP si desea suministrar una dirección única o de comodín. wildcard address. Por ejemplo, ADDRESS('192.168.*') bloquea las conexiones procedentes de una dirección IP que empieza por 192.168.

  Para obtener más información sobre cómo filtrar direcciones IP con patrones, consulte Direcciones IP genéricas.
- TYPE(BLOCKADDR) - Utilice BLOCKADDR si desea proporcionar una lista de direcciones a bloquear.

Parámetros adicionales
Estos parámetros dependen del tipo de regla utilizada en la segunda parte del mandato:

- Para TYPE(ADDRESSMAP), se utiliza ADDRESS
- Para TYPE(BLOCKADDR), se utiliza ADDRLIST

Referencia relacionada
SET CHLAUTH

Bloqueo temporal de direcciones IP específicas si el gestor de colas no está en ejecución
Es posible que quiera bloquear direcciones IP específicas, o rangos de direcciones, cuando el gestor de colas no se esté ejecutando y, por lo tanto, no pueda emitir mandatos MQSC. Puede bloquear temporalmente direcciones IP de forma excepcional modificando el archivo `blockaddr.ini`.

Acerca de esta tarea

El archivo `blockaddr.ini` contiene una copia de las definiciones BLOCKADDR que utiliza el gestor de colas. El escucha lee este archivo si se inicia antes que el gestor de colas. En estas circunstancias, el escucha utiliza los valores añadidos manualmente al archivo `blockaddr.ini`.

No obstante, tenga en cuenta que, cuando el gestor de colas se inicia, graba el conjunto de definiciones BLOCKADDR en el archivo `blockaddr.ini`, sobrescribiendo cualquier edición manual que se haya realizado. De forma similar, cada vez que se añade o se suprime una definición BLOCKADDR mediante el mandato **SET CHLAUTH**, el archivo `blockaddr.ini` se actualiza. Por lo tanto, puede realizar cambios permanentes en las definiciones BLOCKADDR sólo mediante el mandato **SET CHLAUTH** cuando el gestor de colas se esté ejecutando.

Procedimiento

1. Abra el archivo `blockaddr.ini` en un editor de texto.

   El archivo se encuentra en el directorio de datos del gestor de colas.
2. Añada direcciones IP como simples pares de palabra clave-valor, donde la palabra clave es `Addr`.

   Si desea información sobre cómo filtrar direcciones IP con patrones, consulte Direcciones IP genéricas.

   Por ejemplo:

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

Tareas relacionadas

“Bloquear direcciones IP específicas” en la página 391
Puede impedir que un canal específico acepte una conexión entrante de una dirección IP o impedir que el gestor de colas en su conjunto permita el acceso desde una dirección IP, utilizando un registro de autenticación de canal.

Referencia relacionada

SET CHLAUTH

Bloquear identificadores (ID) de usuario específicos

Puede impedir que usuarios específicos utilicen un canal, especificando identificadores (ID) de usuario que, si se confirman, hacen que el canal finalice. Para ello, establezca un registro de autenticación de canal.

Antes de empezar

Asegúrese de que los registros de canal de autenticación estén habilitados del modo siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procedimiento

Establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record**. Por ejemplo, puede emitir el mandato MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.

La lista de usuarios proporcionada en un TYPE(BLOCKUSER) sólo se aplica a los canales SVRCONN y no a los canales del gestor de colas al gestor de colas.

IDusuario1 e *IDusuario2* son cada uno el ID de un usuario al que se va a impedir utilizar el canal. También puede también especificar el valor especial `*MQADMIN` para hacer referencia a los usuarios con privilegios administrativos. Para obtener más información acerca de los usuarios privilegiados, consulte “Usuarios privilegiados” en la página 326. Para obtener más información sobre `*MQADMIN`, consulte SET CHLAUTH.

Referencia relacionada
SET CHLAUTH

Correlacionar un gestor de colas remoto con un ID de usuario MCAUSER
Puede utilizar un registro de autenticación de canal para establecer el atributo MCAUSER de un canal, según el gestor de colas desde el que se conecta el canal.

Antes de empezar
Asegúrese de que los registros de canal de autenticación estén habilitados del modo siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Acerca de esta tarea

Opcionalmente, puede restringir las direcciones IP a las que se aplica la regla.

Tenga en cuenta que esta técnica no se aplica a canales de conexión con el servidor. Si especifica el nombre de un canal de conexión con el servidor en los mandatos siguientes, no tiene ningún efecto.

Procedimiento

- Establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record**. Por ejemplo, puede emitir el mandato MQSC:

  ```
  SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
  ) USERSRC(MAP) MCAUSER(user)
  ```

  nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.
  nombre-gestcolas-asociado-genérico es el nombre del gestor de colas, o un patrón que incluye el símbolo de asterisco (*) como comodín que coincide con el nombre del gestor de colas.
  usuario es el ID de usuario que se utilizará para todas las conexiones del gestor de colas especificado.
- Para restringir este mandato a determinadas direcciones IP, incluya el parámetro **ADDRESS**, de la siguiente manera:

  ```
  SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
  ) USERSRC(MAP) MCAUSER(user) ADDRESS(
  generic-ip-address)
  ```

  nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.
  dirección-ip-genérica es una dirección individual, o un patrón que incluye el símbolo asterisco (*) como comodín o el guión (-) para indicar un rango, que coincide con la dirección. Si desea más información sobre direcciones IP genéricas, consulte Direcciones IP genéricas.

Referencia relacionada
SET CHLAUTH

Correlación de un ID de usuario cliente con un ID de usuario MCAUSER
Se puede utilizar un registro de autenticación de canal para cambiar el atributo MCAUSER de un canal de conexión con el servidor en función del ID de usuario recibido de un cliente.

Antes de empezar
Asegúrese de que los registros de canal de autenticación estén habilitados del modo siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Acerca de esta tarea

Tenga en cuenta que esta técnica sólo se aplica a canales de conexión con el servidor. No tiene ningún efecto en otros tipos de canal.

Procedimiento

Establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record** . Por ejemplo, puede emitir el mandato MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
MCAUSER(
user)
```

nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.

nombre-usuario-cliente es el ID de usuario asociado a la conexión de cliente, el valor podría ser confirmado por la aplicación cliente, modificado por autenticación de conexión usando una adopción temprana o establecido vía salida de canal.

usuario es el ID de usuario que se utilizará en lugar del nombre de usuario del cliente.

Referencia relacionada

SET CHLAUTH

Atributos de la stanza de canales (ChlauthEarlyAdopt)

Correlacionar un Nombre distinguido SSL o TLS con un ID de usuario MCAUSER

Puede utilizar un registro de autenticación de canal para establecer el atributo MCAUSER de un canal, según el Nombre distinguido (DN) recibido.

Antes de empezar

Asegúrese de que los registros de canal de autenticación estén habilitados del modo siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procedimiento

Establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record**. Por ejemplo, puede emitir el mandato MQSC:

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP)
SSLPEER(generic-ssl-peer-name) SSLCERTI(generic-issuer-name)
USERSRC(MAP) MCAUSER(user)
```

nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.

nombre-igual-ssl-genérico es una serie que sigue las reglas de IBM MQ estándar para los valores de SSLPEER. Consulte Reglas de IBM MQ para valores SSLPEER.

usuario es el ID de usuario que se utilizará para todas las conexiones que utilicen el DN especificado.

nombre-emisor-genérico hace referencia al DN del emisor del certificado que ha de coincidir. Este parámetro opcional pero debe utilizarlo para evitar que el certificado erróneo coincida falsamente si se utilizan varias entidades emisoras de certificados.

Referencia relacionada

SET CHLAUTH

Bloquear el acceso desde un gestor de colas remoto
Puede utilizar un registro de autenticación de canal para impedir que un gestor de colas remoto inicie canales.

Antes de empezar
Asegúrese de que los registros de canal de autenticación estén habilitados del modo siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Acerca de esta tarea
Tenga en cuenta que esta técnica no se aplica a canales de conexión con el servidor. Si especifica el nombre de un canal de conexión con el servidor en el mandato siguiente, no tiene ningún efecto.

Procedimiento
Establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record**. Por ejemplo, puede emitir el mandato MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(QMGRMAP) QMNAME(' generic-partner-qmgr-name ')
USERSRC(NOACCESS)
```

nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.
nombre-gestcolas-asociado-genérico es el nombre del gestor de colas, o un patrón que incluye el símbolo de asterisco (*) como comodín que coincide con el nombre del gestor de colas.

Referencia relacionada
SET CHLAUTH

Bloqueo del acceso de un ID de usuario cliente
Se puede utilizar un registro de autenticación de canal para impedir que un ID de usuario establezca una conexión de canal.

Antes de empezar
Asegúrese de que los registros de canal de autenticación estén habilitados del modo siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Acerca de esta tarea
Tenga en cuenta que esta técnica sólo se aplica a canales de conexión con el servidor. No tiene ningún efecto en otros tipos de canal.

Procedimiento
Establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record**. Por ejemplo, puede emitir el mandato MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(USERMAP) CLNTUSER(' client-user-name ')
USERSRC(NOACCESS)
```

nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.

nombre-usuario-cliente es el ID de usuario asociado a la conexión de cliente, el valor podría ser confirmado por la aplicación cliente, modificado por autenticación de conexión usando una adopción temprana o establecido vía salida de canal.

Referencia relacionada

SET CHLAUTH

Bloquear el acceso para un Nombre distinguido SSL o TLS

Puede utilizar un registro de autenticación de canal para impedir que un Nombre distinguido (DN) TLS inicie canales.

Antes de empezar

Asegúrese de que los registros de canal de autenticación estén habilitados del modo siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procedimiento

Establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record**. Por ejemplo, puede emitir el mandato MQSC:

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP)
SSLPEER('generic-ssl-peer-name') SSLCERTI(generic-issuer-name)
USERSRC(NOACCESS)
```

nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.

nombre-igual-ssl-genérico es una serie que sigue las reglas de IBM MQ estándar para los valores de SSLPEER. Consulte Reglas de IBM MQ para valores SSLPEER.

nombre-emisor-genérico hace referencia al DN del emisor del certificado que ha de coincidir. Este parámetro opcional pero debe utilizarlo para evitar que el certificado erróneo coincida falsamente si se utilizan varias entidades emisoras de certificados.

Referencia relacionada

SET CHLAUTH

Correlacionar una dirección IP con un ID de usuario MCAUSER

Puede utilizar un registro de autenticación de canal para establecer el atributo MCAUSER de un canal, según la dirección IP desde la que se recibe la conexión.

Antes de empezar

Asegúrese de que los registros de canal de autenticación estén habilitados del modo siguiente:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Procedimiento

Establezca un registro de autenticación de canal utilizando el mandato MQSC **SET CHLAUTH**o el mandato PCF **Set Channel Authentication Record**. Por ejemplo, puede emitir el mandato MQSC:

```
SET CHLAUTH(' generic-channel-name ') TYPE(ADDRESSMAP) ADDRESS(' generic-ip-address ')
USERSRC(MAP) MCAUSER(user)
```

nombre-canal-genérico es el nombre de un canal para el que desea controlar el acceso, o un patrón que incluye el símbolo asterisco (*) como un comodín que coincide con el nombre de canal.

usuario es el ID de usuario que se utilizará para todas las conexiones que utilicen el DN especificado.

dirección-ip-genérica es la dirección desde la que se establece la conexión, o un patrón que incluye el asterisco (*) como comodín o el guión (-) para indicar un rango, que coincide con la dirección.

Referencia relacionada

SET CHLAUTH

Inhabilitar el acceso remoto al gestor de colas

Si no desea que las aplicaciones cliente se conecten con su gestor de colas, inhabilite el acceso remoto a ellas.

Acerca de esta tarea

Evite que las aplicaciones clientes se conecten al gestor de colas de una de las maneras siguientes:

Procedimiento

- Suprima todos los canales de conexión con el servidor utilizando el mandato MQSC **DELETE CHANNEL**.
- Establezca como identificador de usuario del agente del canal de mensajes (MCAUSER) del canal un ID de usuario sin derecho de acceso, mediante el mandato MQSC **ALTER CHANNEL**.

Configurar la seguridad de conexión

Otorgue la autorización para conectarse con el gestor de colas a cada usuario o grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para configurar la seguridad de conexión, utilice los mandatos adecuados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  En AIX, Linux, and Windows:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  ```

- IBM i
  En IBM i:

  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
  ```

- z/OS
  En 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)
  ```

  Estos mandatos otorgan autorización de conexión para el lote, CICS, IMS y el iniciador de canal (CHIN). Si no utiliza un tipo concreto de conexión, omita los mandatos correspondientes.

  Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

ObjectProfile
El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

Conceptos relacionados

"Connection security profiles for the channel initiator" en la página 207
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.

Control del acceso de los usuarios a las colas

Desea controlar el acceso de la aplicación a las colas. Utilice este tema para determinar qué acciones deben llevarse a cabo.

Para cada declaración true de la primera columna, lleve a cabo la acción indicada en la segunda columna.

Sentencia	Acción
La aplicación obtiene menajes de una cola	Consulte "Otorgar autorización para obtener mensajes de colas" en la página 399.
La aplicación establece el contenido	Consulte "Otorgar autorización para establecer contexto" en la página 400.
La aplicación pasa el contexto	Consulte "Otorgar autorización para pasar contexto" en la página 401.
La aplicación transfiere mensajes a una cola agrupada en clúster	Consulte "Autorización de transferencia de mensajes a colas de clústeres remotos" en la página 487.
La aplicación transfiere mensajes a una cola local	Consulte "Otorgar autorización para transferir mensajes a una cola local" en la página 402.
La aplicación transfiere mensajes a una cola modelo	Consulte "Otorgar autorización para transferir mensajes a una cola modelo" en la página 403.
La aplicación transfiere mensajes a una cola remota	Consulte "Otorgar autorización para transferir mensajes a una cola de clúster remota" en la página 403.

Otorgar autorización para obtener mensajes de colas
Otorgue la autorización para obtener mensajes de una cola o un conjunto de colas a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para obtener mensajes de algunas colas locales, utilice los mandatos adecuados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- Windows
  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- IBM i

  Para IBM i, emita el mandato siguiente:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME(' QMgrName ')
  ```

- z/OS

  Para z/OS, emita los mandatos siguientes:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor también puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para establecer contexto
Otorgue la autorización para establecer contexto en un mensaje recibido que se está transfiriendo a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para establecer contexto en algunas colas, utilice los mandatos adecuados de para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  Para sistemas AIX, Linux, and Windows, emita uno de los mandatos siguientes:

  - Para establecer sólo contexto de identidad:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
    ```

  - Para establecer todo el contexto:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setall
    ```

  Nota: Para utilizar las autoridades `setid` o `setall`, hay que otorgar las autoridades al correspondiente objeto de cola y también al objeto de gestor de colas.

- IBM i

  Para IBM i, emita uno de los siguientes mandatos:

  - Para establecer sólo contexto de identidad:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME('
QMgrName ')
```

- Para establecer todo el contexto:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('
QMgrName ')
```

- z/OS

  Para z/OS, emita uno de los siguientes conjuntos de mandatos:

  - Para establecer sólo contexto de identidad:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  - Para establecer todo el contexto:

```
RDEFINE MQQUEUE QMgrName. ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para pasar contexto
Otorgue la autorización para pasar contexto de un mensaje recibido a uno que se está transfiriendo a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para pasar contexto a algunas colas, utilice los mandatos adecuados de para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  Para sistemas AIX, Linux, and Windows, emita uno de los mandatos siguientes:

  - Para pasar sólo contexto de identidad:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
```

  - Para pasar todo el contexto:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
```

- IBM i

  Para IBM i, emita uno de los siguientes mandatos:

- Para pasar sólo contexto de identidad:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('
QMgrName ')
```

- Para pasar todo el contexto:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('
QMgrName ')
```

- z/OS

  Para z/OS, emita los siguientes mandatos para pasar contexto de identidad o todo el contexto:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para transferir mensajes a una cola local
Otorgue la autorización para transferir mensajes a una cola local o a un conjunto de colas a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para transferir mensajes a algunas colas locales, utilice los mandatos adecuados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM i

  Para IBM i, emita el mandato siguiente:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- z/OS

  Para z/OS, emita los mandatos siguientes:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

ObjectProfile
El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para transferir mensajes a una cola modelo
Otorgue la autorización para transferir mensajes a una cola modelo o a un conjunto de colas modelo a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Las colas modelo se utilizan para crear colas dinámicas. Por lo tanto, debe otorgar autorización a la colas modelo y dinámicas. Para otorgar estas autorizaciones, utilice los mandatos adecuados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  Para sistemas AIX, Linux, and Windows, emita los mandatos siguientes:

```
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 los mandatos siguientes:

```
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 los mandatos siguientes:

```
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)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  NombreColaModelo
  El nombre de la cola modelo en la que se basan las colas dinámicas.

  ObjectProfile
  El nombre de la cola dinámica o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para transferir mensajes a una cola de clúster remota
Otorgue la autorización para transferir mensajes a una cola de clúster remota o a un conjunto de colas a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para colocar un mensaje en una cola de clúster remota puede ponerlo en una definición local de una cola remota o en un cola remota con nombre completo. Si está utilizando una definición local de una cola remota, necesitará la autoridad para transferir el objeto local: consulte “Otorgar autorización para transferir mensajes a una cola local” en la página 402. Si utiliza un completo de la cola remota, necesitará autorización para transferir a la cola remota. Otorgue esta autorización mediante los mandatos adecuados para su sistema operativo.

El comportamiento predeterminado es realizar el control de acceso para `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Tenga en cuenta que este comportamiento se aplica, incluso si está utilizando varias colas de transmisión.

El comportamiento descrito en este tema solamente se aplica si ha configurado el atributo **`ClusterQueueAccessControl`** en el archivo `qm.ini` para que sea *RQMName*, tal como se describe en el tema Stanza de seguridad y si ha reiniciado el gestor de colas.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

  ```
  setmqaut -m QMgrName -t rqmname -n
  ObjectProfile -g GroupName +put
  ```

  Tenga en cuenta que puede utilizar el objeto *rqmname* para las colas de clúster remoto sólo.
- IBM i
  Para IBM i, emita el mandato siguiente:

  ```
  GRTMQMAUT OBJTYPE(*RMTMQMNAME) OBJ('
  ObjectProfile') USER(GroupName) AUT(*PUT) MQMNAME('
  QMgrName')
  ```

  Tenga en cuenta que puede utilizar el objeto RMTMQMNAME para las colas de clúster remoto sólo.
- z/OS
  Para z/OS, emita los mandatos siguientes:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  Tenga en cuenta que puede usar el nombre del gestor de colas remoto (o el grupo de compartición de colas) solo para las colas de clúster remoto.

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del gestor de colas remoto o el perfil genérico para el cual se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Control del acceso de los usuarios a los temas

Necesita controlar el acceso de las aplicaciones a los temas. Utilice este tema para determinar qué acciones deben llevarse a cabo.

Para cada declaración true de la primera columna, lleve a cabo la acción indicada en la segunda columna.

Tabla 74. Control del acceso de los usuarios a los temas

Sentencia	**Acción**
La aplicación publica mensajes en un tema	Consulte "Otorgar autorización para publicar mensajes en un tema" en la página 405.
La aplicación se suscribe a un tema	Consulte "Otorgar autorización para suscribirse a temas" en la página 405.

Otorgar autorización para publicar mensajes en un tema

Otorgue la autorización para publicar mensajes en un tema o un conjunto de temas a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para publicar mensajes en algunos temas, utilice los mandatos adecuados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
  ```

- IBM i

  Para IBM i, emita el mandato siguiente:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('
  QMgrName ')
  ```

- z/OS

  Para z/OS, emita los mandatos siguientes:

  ```
  RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para suscribirse a temas

Otorgue la autorización para acceder a un tema o un conjunto de temas a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para suscribirse a algunos temas, utilice los mandatos adecuados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
  ```

- IBM i

  Para IBM i, emita el mandato siguiente:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('
  QMgrName ')
  ```

- z/OS

  Para z/OS, emita los mandatos siguientes:

  ```
  RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
  PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para consultar en un gestor de colas

Otorgue la autorización para consultar en un gestor de colas en cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para consultar en un gestor de colas, utilice los mandatos adecuados de para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
  ```

- IBM i

  Para IBM i, emita el mandato siguiente:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME(' QMgrName
')
```

- z/OS

  Para z/OS, emita los mandatos siguientes:

  ```
  RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Estos mandatos otorgan acceso al gestor de cola especificado. Para permitir al usuario utilizar el mandato MQINQ, emita los mandatos siguientes:

  ```
  RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
  PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para acceder a procesos

Otorgue la autorización para acceder a un proceso o un conjunto de procesos a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para acceder a algunos procesos, utilice los mandatos adecuados de para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW

  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
  ```

- IBM i

  Para IBM i, emita el mandato siguiente:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME(' QMgrName
  ')
  ```

- z/OS

  Para z/OS, emita los mandatos siguientes:

  ```
  RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
  ```

  Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

ObjectProfile
El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

Otorgar autorización para acceder a listas de nombres

Otorgue la autorización para acceder a una lista de nombres o un conjunto de listas de nombres a cada grupo de usuarios con una necesidad empresarial para ello.

Acerca de esta tarea

Para otorgar autorización para acceder a algunas listas de nombres, utilice los mandatos adecuados para su sistema operativo.

Multi En Multiplatforms, también puede utilizar el mandato SET AUTHREC .

Procedimiento

- ALW
  Para sistemas AIX, Linux, and Windows, emita el mandato siguiente:

```
setmqaut -m QMgrName -n
ObjectProfile -t namelist -g GroupName
+all
```

- IBM i
  Para IBM i, emita el mandato siguiente:

```
GRTMQMAUT OBJ('ObjectProfile
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- z/OS
  Para z/OS, emita los mandatos siguientes:

```
RDEFINE MQNLIST
QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile
CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
```

  Los nombres de las variables tienen los significados siguientes:

  QMgrName
  Nombre del gestor de colas. En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.

  ObjectProfile
  El nombre del objeto o el perfil genérico para el que se van a cambiar autorizaciones.

  GroupName
  Nombre del grupo al que se va a otorgar el acceso.

ALW Autorización para administrar IBM MQ en AIX, Linux, and Windows

Los administradores de IBM MQ pueden utilizar todos los mandatos de IBM MQ y otorgar autorizaciones para otros usuarios. Cuando los administradores emiten mandatos a gestores de colas remotos, deben tener la autorización necesaria en el gestor de colas remoto. Se aplican otras Windows.

Los administradores de IBM MQ tienen autorización para utilizar todos los mandatos de IBM MQ (incluidos los mandatos para otorgar autorizaciones de IBM MQ a otros usuarios).

Para ser administrador de IBM MQ, hay que ser miembro de un grupo especial llamado grupo **mqm**.

Windows De forma alternativa, solo en Windows, las cuentas locales pueden administrar IBM MQ si son miembros del grupo de administradores en los sistemas Windows.

Atención: puede añadir el usuario de Azure AD al grupo mqm utilizando un mandato de administrador. Por ejemplo, utilice el mandato `net localgroup mqm AzureAD\<your userID> /add`. A continuación, ejecute los mandatos de administración de IBM MQ o utilice IBM MQ Explorer.

El grupo **mqm** se crea automáticamente cuando se instala IBM MQ. Se pueden añadir más usuarios al grupo para permitirles realizar tareas de administración. Todos los miembros de este grupo tienen acceso a todos los recursos. Este acceso solo se puede revocar eliminando un usuario del grupo **mqm** y ejecutando el mandato **REFRESH SECURITY**.

Los administradores pueden utilizar los mandatos de control para administrar IBM MQ. Uno de estos mandatos de control es **setmqaut**, que se utiliza para conceder autorización a otros usuarios para que puedan acceder a los recursos de IBM MQ o controlarlos. Los mandatos para gestionar registros de autorización PCF están a la disposición de aquellos usuarios que no son administradores a quienes se les han otorgado las autorizaciones dsp y chg en el gestor de colas. Para obtener más información sobre la gestión de las autorizaciones con mandatos PCF, consulte Formatos de mandatos programables.

Los administradores deben tener las autorizaciones necesarias para que el gestor de colas remoto procese los mandatos MQSC. IBM MQ Explorer emite mandatos PCF para realizar tareas de administración. Los administradores no necesitan autorizaciones adicionales para utilizar IBM MQ Explorer para administrar un gestor de colas en el sistema local. Cuando IBM MQ Explorer se utiliza para administrar un gestor de colas en otro sistema, los administradores deben tener las autorizaciones necesarias para que el gestor de colas remoto procese los mandatos PCF.

Atención: No es necesario que sea un administrador para utilizar el mandato de control **runmqsc**, que emite mandatos de script de IBM MQ (MQSC).

Cuando se utiliza **runmqsc** en modalidad indirecta para enviar mandatos MQSC a un gestor de colas remoto, todo mandato MQSC se encapsula en un mandato PCF de escape.

Para obtener más información acerca de las comprobaciones de autorización cuando se procesan los mandatos PCF y MQSC, consulte los temas siguientes:

- Para los mandatos PCF que operan en los gestores de colas, colas, procesos, listas de nombres y objetos de información de autenticación, consulte Autorización para trabajar con objetos de IBM MQ. Consulte en este apartado los mandatos MQSC equivalentes encapsulados en mandatos PCF de escape.
- Para los mandatos PCF que se ejecutan en canales, iniciadores de canal, escuchas y clústeres, consulte Seguridad de canal.
- Para los mandatos PCF que operan en los registros de autorización, consulte Comprobación de autorización para mandatos EN PCF
- z/OS Para los mandatos MQSC que procesa el servidor de mandatos en IBM MQ for z/OS, consulte Seguridad de mandatos y seguridad de recursos de mandatos en z/OS .

Además, en sistemas Windows , la cuenta SYSTEM tiene acceso completo a los recursos de IBM MQ .

En las plataformas AIX and Linux, también se crea un ID de usuario de **mqm** para uso exclusivo del producto. Este ID no debe estar disponible nunca para los usuarios que no tienen estos privilegios. Todos los objetos de IBM MQ son propiedad del ID de usuario de **mqm**.

En sistemas Windows , los miembros del grupo Administradores también pueden administrar cualquier gestor de colas, al igual que la cuenta SYSTEM. También puede crear un grupo de dominio **mqm** en el controlador de dominios que contenga todos los ID de usuario con privilegios que están activos en el dominio y añadirlo al grupo **mqm** local. Algunos mandatos, por ejemplo **crtmqm**, manipulan autorizaciones

sobre objetos de IBM MQ y por ello necesitan autorización para trabajar con estos objetos (tal como se describe en los apartados siguientes). Los miembros del grupo **mqm** tienen autorización para trabajar con todos los objetos, pero en sistemas Windows se puede dar el caso en que se deniegue la autorización si hay un usuario local y un usuario autenticado por el dominio con el mismo nombre. Este tema se describe en el apartado "Principales y grupos en AIX, Linux, and Windows" en la página 413.

Las versiones de Windows con una característica de Control de cuentas de usuario (UAC) restringe las acciones que los usuarios pueden llevar a cabo en determinados recursos del sistema operativo, incluso si son miembros del grupo Administradores. Si su ID de usuario está en el grupo Administradores pero no en el grupo **mqm**, hay que utilizar un indicador de mandatos elevado para ejecutar mandatos de administración de IBM MQ como, por ejemplo, **crtmqm**; de lo contrario se genera el error `"AMQ7077: No tiene autorización para realizar la operación solicitada"`. Para abrir un indicador de mandatos elevado, pulse el botón derecho del ratón en el elemento de menú, o icono, de inicio, para el indicador de mandatos, y seleccione **Ejecutar como administrador**.

No es necesario ser miembro del grupo **mqm** para realizar las acciones siguientes:

- Emitir mandatos desde un programa de aplicación que emite mandatos PCF, o mandatos MQSC dentro de un mandato PCF de escape, a menos que los mandatos manipulen iniciadores de canal. (Estos mandatos se describen en "Protección de las definiciones de iniciador de canal" en la página 122).
- Emitir llamadas MQI desde un programa de aplicación (a menos que desee utilizar los enlaces de vía rápida en la llamada `MQCONNX`).
- Utilice el mandato `crtmqcvx` para crear un fragmento de código que realice la conversión de datos en estructuras de tipo de datos.
- Utilizar el mandato `dspmq` para visualizar gestores de colas.
- Utilice el mandato `dspmqtrc` para visualizar la salida de rastreo con formato de IBM MQ.

Se aplica una limitación de 12 caracteres al grupo y a los ID de usuario.

Las plataformas UNIX and Linux suelen restringir la longitud de un ID de usuario a 12 caracteres. AIX 5.3 ha aumentado este límite, pero IBM MQ sigue observando una restricción de 12 caracteres en todas las plataformas UNIX and Linux . Si utiliza un ID de usuario de más de 12 caracteres, IBM MQ lo sustituye por el valor `UNKNOWN`. No defina un ID de usuario con un valor de `UNKNOWN`.

ALW Gestión del grupo mqm en AIX, Linux, and Windows

Se otorgan privilegios administrativos completos a los usuarios del grupo mqm a través de IBM MQ. Por este motivo, no debe inscribir aplicaciones y usuarios ordinarios en el grupo mqm. El grupo mqm sólo debe contener las cuentas de los administradores de IBM MQ.

Estas tareas se describen en el apartado:

- Windows Creación y gestión de grupos en Windows
- AIX Creación y gestión de grupos en AIX
- Linux Creación y gestión de grupos en Linux

Windows Si el controlador de dominio se ejecuta en Windows 2000 o en Windows 2003 o posterior, es posible que el administrador del dominio tenga que establecer una cuenta especial para que la utilice IBM MQ. Para obtener más información, consulte Configuración de IBM MQ con Prepare IBM MQ Wizard y Creación y configuración de cuentas de dominio de Windows para IBM MQ.

ALW Autorización para trabajar con objetos IBM MQ en AIX, Linux, and Windows

Todos los objetos están protegidos por IBM MQ y los principales deben recibir la autorización adecuada para acceder a ellos. Diferentes principales necesitan diferentes derechos de acceso a diferentes objetos.

Se accede a los gestores de colas, colas, definiciones de proceso, listas de nombres, canales, canales de conexión de cliente, escuchas, servicios y objetos de información de autenticación desde aplicaciones que utilizan llamadas MQI o mandatos PCF. Estos recursos están todos protegidos por IBM MQ, y las aplicaciones deben tener permiso para acceder a ellos. La entidad que realiza la solicitud puede ser un usuario, un programa de aplicación que emite una llamada MQI o un programa de administración que emite un mandato PCF. Se hace referencia al identificador del peticionario como *principal*.

Se puede otorgar a distintos grupos de principales diferentes tipos de autorización de acceso al mismo objeto. Por ejemplo, para una cola específica, puede permitirse a un grupo que realice operaciones de transferir y obtener; otro grupo puede tener únicamente autorización para examinar la cola (MQGET con la opción de examinar). Del mismo modo, algunos grupos pueden tener autorización de transferir y obtener para una cola pero pueden no tener autorización para alterar los atributos de la cola o suprimirla.

Algunas operaciones son especialmente comprometidas y deberían limitarse a usuarios con privilegios. Por ejemplo:

- Acceder a algunas colas especiales, tales como las colas de transmisión o la cola de mandatos SYSTEM.ADMIN.COMMAND.QUEUE
- La ejecución de programas que utilicen todas las opciones de contexto de la MQI
- La creación y supresión de colas de aplicación

Se concede automáticamente permiso de acceso completo sobre un objeto al ID de usuario que ha creado el objeto y a todos los miembros del grupo mqm (y también a los miembros del grupo Administradores en sistemas Windows).

Conceptos relacionados

"Autorización para administrar IBM MQ en AIX, Linux, and Windows" en la página 408
Los administradores de IBM MQ pueden utilizar todos los mandatos de IBM MQ y otorgar autorizaciones para otros usuarios. Cuando los administradores emiten mandatos a gestores de colas remotos, deben tener la autorización necesaria en el gestor de colas remoto. Se aplican otras Windows.

ALW Cuándo se realizan comprobaciones de seguridad en AIX, Linux, and Windows

Las comprobaciones de seguridad normalmente se realizan al conectar a un gestor de colas, al abrir o cerrar objetos y al transferir u obtener mensajes.

Las comprobaciones de seguridad que se realizan en una aplicación típica son las siguientes:

Conectar al gestor de colas (llamadas MQCONN o MQCONNX)
Ésta es la primera vez que la aplicación se asocia a un gestor de colas determinado. El gestor de colas investiga en el entorno operativo para detectar el ID de usuario asociado a la aplicación. A continuación, IBM MQ comprueba que el ID de usuario tiene autorización para conectarse con el gestor de colas y guarda el ID de usuario para futuras comprobaciones.

Los usuarios no necesitan iniciar la sesión en IBM MQ; IBM MQ presupone que los usuarios han iniciado la sesión en el sistema operativo y que éste los ha autenticado.

Abrir el objeto (llamadas MQOPEN o MQPUT1)
Se accede a los objetos IBM MQ abriendo el objeto y emitiendo mandatos para el mismo. Todas las comprobaciones de recursos se realizan cuando se abre el objeto, en lugar de hacerlo cuando se accede al mismo. Esto significa que la solicitud **MQOPEN** debe especificar el tipo de acceso necesario (por ejemplo, si el usuario simplemente desea examinar el objeto o realizar una actualización como, por ejemplo, colocar mensajes en una cola).

IBM MQ comprueba el recurso que se nombra en la solicitud **MQOPEN**. En un objeto de cola alias o remota, la autorización que se utiliza es la del objeto propiamente dicho, no la de la cola en la que se resuelve la cola alias o remota. Esto significa que el usuario no necesita tener permiso para acceder al mismo. Limite la autorización para crear colas a los usuarios con privilegios. De otro modo, los usuarios podrán eludir el control de accesos normal simplemente creando un alias. Si se hace

referencia a una cola remota de forma explícita en los nombres de la cola y del gestor de colas, se comprobará la cola de transmisión asociada al gestor de colas remoto.

La autorización sobre una cola dinámica se basa en la cola modelo de la que se deriva, aunque no tiene por qué ser igual. Este tema se describe en la Nota "1" en la página 141.

El ID de usuario que utiliza el gestor de colas para las comprobaciones de acceso es el ID de usuario obtenido desde el sistema operativo de la aplicación conectada al gestor de colas. Una aplicación con las autorizaciones adecuadas puede emitir una llamada **MQOPEN** especificando un ID de usuario alternativo. Las comprobaciones de control de accesos se realizan de este modo en el ID de usuario alternativo. Esto no modifica el ID de usuario asociado a la aplicación, solamente el que se utiliza para las comprobaciones de control de accesos.

Transferir y obtener mensajes (llamadas MQPUT o MQGET)
No se realizan comprobaciones de control de acceso.

Cerrar el objeto (MQCLOSE)
No se realizan comprobaciones de control de accesos a menos que el resultado de la llamada **MQCLOSE** sea la supresión de una cola dinámica. En este caso, se comprueba que el ID de usuario tenga autorización para suprimir la cola.

Suscripción a un tema (MQSUB)
Cuando una aplicación se suscribe a un tema, especifica el tipo de operación que necesita realizar. La operación es crear una nueva suscripción, alterar una suscripción existente o reanudar una suscripción existente sin modificarla. Para cada tipo de operación, el gestor de colas comprueba que el ID de usuario asociado a la aplicación tenga autorización para realizar la operación.

Cuando una aplicación se suscribe a un tema, las comprobaciones de autorización se realizan respecto a objetos de temas que se encuentran en el árbol de temas en el punto o por encima del punto del árbol de temas al que se ha suscrito la aplicación. Las comprobaciones de autorización pueden implicar comprobaciones en más de un objeto de tema.

El ID de usuario que utiliza el gestor de colas para las comprobaciones de autorización es el ID de usuario que ha obtenido del sistema operativo cuando la aplicación se conecta al gestor de colas.

El gestor de colas realiza comprobaciones de autorización en las colas de suscriptores pero no en las colas gestionadas.

ALW Cómo implementa IBM MQ el control de accesos en AIX, Linux, and Windows

IBM MQ utiliza los servicios de seguridad proporcionados por el sistema operativo subyacente mediante el gestor de autorizaciones sobre objetos. IBM MQ proporciona mandatos para crear y mantener listas de control de accesos.

Una interfaz de control de accesos llamada Interfaz del servicio de autorización forma parte de IBM MQ. IBM MQ proporciona una implementación de un gestor de control de accesos (conforme a la Interfaz del servicio de autorización) que se conoce como *gestor de autorizaciones sobre objetos (OAM)*. Este gestor se instala y se activa automáticamente para cada gestor de colas que cree, a menos que especifique lo contrario, como se explica en "Impedir comprobaciones de acceso de seguridad en los sistemas AIX, Linux, and Windows" en la página 371). El OAM puede sustituirse por cualquier componente escrito por el usuario o por terceros que esté en conformidad con la Interfaz del servicio de autorización.

El OAM aprovecha las características de seguridad del sistema operativo subyacente, utilizando los ID de usuario y de grupo del sistema operativo. Los usuarios sólo pueden acceder a los objetos de IBM MQ si tienen la autorización correcta. En el apartado "Control del acceso a objetos mediante el OAM en AIX, Linux, and Windows" en la página 360 se explica cómo conceder y denegar esta autorización.

El OAM mantiene una ACL (Access Control List - Lista de control de accesos) para cada recurso que controla. Los datos de autorización se almacenan en una cola local llamada SYSTEM.AUTH.DATA.QUEUE. El acceso a esta cola está restringido a los usuarios del grupo mqm, y adicionalmente en Windows, a los usuarios del grupo Administradores y a los usuarios que han iniciado sesión con el ID SYSTEM. El acceso de usuarios a la cola no se puede cambiar.

IBM MQ proporciona mandatos para crear y mantener listas de control de accesos. Para obtener más información acerca de estos mandatos, consulte “Control del acceso a objetos mediante el OAM en AIX, Linux, and Windows” en la página 360.

IBM MQ pasa al OAM una solicitud que contiene un principal, un nombre de recurso y un tipo de acceso. El OAM otorga o deniega el acceso basándose en la ACL que mantiene. IBM MQ sigue la decisión adoptada por el OAM; si el OAM no puede tomar una decisión, IBM MQ no permite el acceso.

ALW Identificación del ID de usuario en AIX, Linux, and Windows

El gestor de autorizaciones sobre objetos identifica el principal que está solicitando acceso a un recurso. El ID de usuario utilizado como principal varía según el contexto.

El gestor de autorizaciones sobre objetos (OAM) debe poder identificar quién solicita acceso a un recurso determinado. IBM MQ utiliza el término *principal* para referirse a este identificador. El principal se establece cuando la aplicación se conecta por primera vez al gestor de colas; lo determina el gestor de colas a partir del ID de usuario asociado a la aplicación de conexión. (Si la aplicación emite llamadas XA sin establecer conexión con el gestor de colas, el ID de usuario asociado con la aplicación que emite la llamada xa_open se utiliza para las comprobaciones de autorizaciones que realiza el gestor de colas.)

En sistemas AIX and Linux, las rutinas de autorización comprueban el ID de usuario real (conectado) o el ID de usuario efectivo asociado a la aplicación. El ID de usuario comprobado puede depender del tipo de enlace; para obtener información detallada, consulte Servicios instalables.

IBM MQ propaga el ID de usuario que recibe del sistema en la cabecera de mensaje (la estructura MQMD) de cada mensaje para identificar al usuario. Este identificador forma parte de la información de contexto del mensaje y se describe en el apartado “Autorización de contexto en AIX, Linux, and Windows” en la página 416. Las aplicaciones no pueden alterar esta información a menos que tengan autorización para cambiar la información de contexto.

ALW *Principales y grupos en AIX, Linux, and Windows*

Los principales pueden pertenecer a grupos. Al otorgar el acceso a recursos a grupos en vez de a usuarios individuales, puede reducir la cantidad de administración necesaria. Las listas de control de acceso (ACL) se basan en grupos y en los ID de usuario.

Por ejemplo, puede definir un grupo que conste de usuarios que deseen ejecutar una aplicación determinada. A otros usuarios se les puede permitir el acceso a todos los recursos que necesiten añadiendo su ID de usuario al grupo adecuado.

Este proceso de definir y gestionar grupos se describe para plataformas concretas:

- AIX Creación y gestión de grupos en AIX
- Linux Creación y gestión de grupos en Linux
- Windows Creación y gestión de grupos en Windows

Un principal puede pertenecer a más de un grupo (su conjunto de grupos). Tiene la suma de todas las autorizaciones que se han otorgado a cada grupos en su conjunto de grupos. Estas autorizaciones se almacenan en memoria caché, de este modo los cambios que realice en la pertenencia del grupo del principal no se reconocen, hasta que se reinicia el gestor de colas, a menos que emita el mandato MQSC **REFRESH SECURITY** (o su PCF equivalente).

Linux AIX Sistemas AIX and Linux

Las listas de control de accesos (ACL) se basan en ID de usuario y grupos y puede utilizar para la autorización estableciendo el atributo **SecurityPolicy** en el valor adecuado tal como se describe en Stanza de servicio del archivo qm.ini.

Puede utilizar el *modelo basado en usuario* para la autorización, y esto le permite utilizar tanto usuarios como grupos. Sin embargo, al especificar un usuario en el mandato setmqaut, los nuevos permisos se aplican solo a dicho usuario y no a los grupos a los que pertenece dicho usuario. Para

obtener más información, consulte “Permisos basados en usuario de OAM en AIX and Linux” en la página 360.

Cuando se utiliza el *modelo basado en grupo* para la autorización, el grupo primario al que pertenece el ID de usuario se incluye en la ACL. El ID de usuario individual no se incluye y la autorización se otorga a todos los miembros de dicho grupo. Por eso, tenga en cuenta que puede modificar accidentalmente la autorización de un principal al modificar la autorización de otro principal del mismo grupo.

Todos los usuarios se asignan nominalmente al grupo de usuarios predeterminado `nadie` y de forma predeterminada, a este grupo no se le concede ningún tipo de autorización. Puede cambiar la autorización del grupo `nadie` para otorgar acceso a los recursos IBM MQ a aquellos usuarios que no tienen autorizaciones específicas.

DeIBM MQ 9.3.0 , puedes usar el`UserExternal` opción de la**SecurityPolicy** atributo para crear un nombre de usuario que no sea del sistema operativo. Si crea un nombre de usuario que no sea del sistema operativo, se considera que ese usuario no pertenece a ningún grupo, excepto elnobody grupo. Para obtener más información sobre esta opción, consulte crtmqm y Estrofa de servicio delqm.ini archivo .

No defina un ID de usuario con el valor `UNKNOWN`. El valor `UNKNOWN` se utiliza cuando ID de usuario es demasiado largo, por lo que los ID de usuario arbitrarios deberían utilizar las autorizaciones de acceso de `UNKNOWN`.

Consulte “Configuración de autorizaciones” en la página 422 para obtener información sobre cómo utilizar LDAP.

Los ID de usuario tener 12 caracteres como máximo y los nombres de grupo también.

Windows Sistemas Windows

Las ACL están basadas en los grupos y en los ID de usuario. Las comprobaciones son las mismas que para AIX and Linux. Puede tener usuarios diferentes en dominios diferentes con el mismo ID de usuario. IBM MQ permite que los ID de usuario se califiquen con el nombre de dominio, de modo que estos usuarios puedan tener diferentes niveles de acceso.

El nombre del grupo puede incluir opcionalmente un nombre de dominio, especificado con los formatos siguientes:

```
GroupName@domain domain_name\group_name
```

Los grupos globales son comprobados por el OAM sólo en dos casos:

1. La stanza de seguridad del gestor de colas incluye el valor: `GroupModel=GlobalGroups`. Consulte Seguridad.
2. El gestor de colas está utilizando un grupo de acceso de seguridad alternativo. Consulte **crtmqm** .

Los ID de usuario pueden tener hasta 20 caracteres, los nombres de dominio hasta 15 caracteres y los nombres de grupo hasta 64 caracteres.

El OAM comprueba en primer lugar la base de datos de seguridad local, luego la base de datos del dominio primario y, finalmente, la base de datos de cualquier dominio fiable. Para la comprobación, el OAM utiliza el primer ID de usuario que encuentra. Cada uno de estos ID de usuario puede tener distintos miembros de grupos en un sistema determinado.

Algunos mandatos de control (por ejemplo, **crtmqm**) modifican las autorizaciones sobre objetos IBM MQ utilizando el gestor de autorizaciones sobre objetos (OAM). El OAM busca en las bases de datos de seguridad en el orden dado para determinar los derechos de autorización de un ID de usuario específico. La autorización que determine el OAM puede alterar temporalmente el que un ID de usuario sea miembro del grupo mqm local. Por ejemplo, si emite el mandato **crtmqm** desde un ID de usuario autenticado por un controlador de dominio que sea miembro del grupo mqm local a través de un grupo global, el mandato no se ejecutará correctamente si el sistema tiene un usuario local con el mismo nombre que no esté en el grupo mqm local.

Para obtener más información sobre cómo establecer el atributo **SecurityPolicy** en Windows, consulte Stanza de servicio del archivo qm.ini.

Windows *Identificadores de seguridad (SID) de Windows*

IBM MQ en Windows utiliza el identificador de seguridad (SID) cuando está disponible. Si no se proporciona un SID de Windows con una solicitud de autorización, IBM MQ identifica el usuario basándose solamente en el nombre de usuario, pero esto puede dar como resultado que se otorgue la autorización incorrecta.

En sistemas Windows, el identificador de seguridad (SID) se utiliza para complementar el ID de usuario. El SID contiene información que identifica todos los detalles de la cuenta del usuario en la base de datos del administrador de cuentas de seguridad (SAM) de Windows donde se ha definido el usuario. Cuando se crea un mensaje en IBM MQ for Windows, IBM MQ almacena el SID en el descriptor de mensaje. Cuando IBM MQ en Windows realiza comprobaciones de autorización, utiliza el SID para consultar la información completa en la base de datos SAM. Para que esta consulta se lleve a cabo correctamente, la base de datos SAM en la que está definido el usuario debe estar accesible.

De forma predeterminada, si no se proporciona un SID de Windows con una solicitud de autorización, IBM MQ identifica el usuario basándose solamente en el nombre de usuario. Esto se efectúa buscando en las bases de datos de seguridad en el orden siguiente:

1. La base de datos de seguridad local.
2. La base de datos de seguridad del dominio primario.
3. La base de datos de seguridad de dominios fiables.

Si el nombre de usuario no es exclusivo, se puede otorgar una autorización IBM MQ incorrecta. Para evitar este problema, incluya un SID en cada solicitud de autorización; IBM MQ utiliza el SID para establecer las credenciales de usuario.

Para especificar que todas las peticiones de autorización deben incluir un SID, utilice **regedit**. Establezca SecurityPolicy en NTSIDsRequired.

ALW Autorización de usuario alternativo en AIX, Linux, and Windows

Puede especificar que un ID de usuario puede utilizar la autorización de otro usuario cuando accede a un IBM MQ. Esto se denomina *autorización de usuario alternativo* y puede utilizarla en cualquier objeto IBM MQ MQ.

La autorización de usuario alternativo es esencial cuando un servidor recibe peticiones de un programa y desea asegurarse de que el programa tiene la autorización necesaria para la solicitud. El servidor puede tener la autorización necesaria, pero necesita saber si el programa tiene autorización para las acciones que ha solicitado.

Por ejemplo, suponga que un programa servidor que se está ejecutando bajo el ID PAYSERV recupera de una cola un mensaje de solicitud que había transferido a la cola el ID de usuario USER1. Cuando el programa servidor obtiene el mensaje de solicitud, procesa la solicitud y vuelve a transferir la respuesta a la cola de respuestas especificada con el mensaje de solicitud. En lugar de utilizar su propio ID de usuario (PAYSERV) para autorizar la apertura de una cola de respuestas, el servidor puede especificar otro ID de usuario, en este caso, USER1. En este ejemplo, puede utilizar la autorización de usuario alternativo para controlar si PAYSERV puede especificar USER1 como ID de usuario alternativo al abrir la cola de respuestas.

El ID de usuario alternativo se especifica en el campo **AlternateUserId** del descriptor de objeto.

Linux Resolución de determinados problemas de pertenencia a grupos en Linux

Algunos sistemas tardan en devolver información de grupo a través de la serie normal de llamadas de API del sistema operativo **getgrent** y, si la empresa tiene miles de grupos en los que buscar, buscando

en qué grupos se encuentra el usuario `mqm` , la respuesta lenta puede provocar un tiempo de espera de gestor de colas interno. Para evitar este problema, existe una API de sistema operativo alternativa.

Para utilizar la API alternativa que es más rápida y devuelve todos los grupos de una llamada, establezca la variable de entorno MQS_GETGROUPLIST_API.

Es posible que haya recibido un error RC2035 al otorgar acceso de conexión al grupo secundario del usuario y habilitar la variable MQS_GETGROUPLIST_API alivia el problema.

A continuación, IBM MQ utiliza la API **`getgrouplist`** en lugar de la API **`getgrent`** .

Para habilitar **`getgrouplist`**:

1. Detener el gestor de colas
2. Emita el mandato export MQS_GETGROUPLIST_API=1
3. Reinicie el gestor de colas

Vuelva a intentar el escenario que ha fallado y, si el problema se ha resuelto, podría considerar la posibilidad de modificar el archivo `.bashrc / .profile` para el usuario `mqm` para añadir esta variable de entorno, o añadir la variable de entorno en el script que utiliza para iniciar el gestor de colas.

Si el sistema fusiona la información de usuario o grupo para el sistema operativo desde varios repositorios como, por ejemplo, NIS o LDAP, asegúrese de que el grupo o ID de usuario sea coherente en todos los repositorios, incluido el local, ya que se utilizan para instalar y establecer permisos de nivel de sistema operativo.

ALW Autorización de contexto en AIX, Linux, and Windows

El contexto es la información que se aplica a un mensaje determinado y está contenida en el descriptor de mensaje, MQMD, que forma parte del mensaje. Las aplicaciones pueden especificar los datos de contexto cuando se realiza una llamada `MQOPEN` o `MQPUT`.

En la información de contexto hay dos secciones:

Sección de identidad
De quién procede el mensaje. Consta de los campos `UserIdentifier`, `AccountingToken`y `ApplIdentityData` .

Sección de origen
De dónde procede el mensaje y cuándo se ha transferido a la cola. Consta de los campos `PutApplType`, `PutApplName`, `PutDate`, `PutTime`y `ApplOriginData` .

Las aplicaciones pueden especificar los datos de contexto cuando se realiza una llamada `MQOPEN` o `MQPUT`. Estos datos pueden haber sido generados por la aplicación, transmitidos desde otro mensaje o generados por el gestor de colas predeterminado. Por ejemplo, los programas servidor pueden utilizar los datos de contexto para comprobar la identidad del peticionario, con lo que comprueban si el mensaje procede de una aplicación que se ejecuta bajo un ID de usuario autorizado.

Un programa servidor puede utilizar el campo `UserIdentifier` para determinar el ID de usuario de un usuario alternativo. La autorización de contexto se utiliza para controlar si el usuario puede especificar alguna de las opciones de contexto en cualquier `MQOPEN` o `MQPUT1`.

Consulte Control de información de contexto para obtener información sobre las opciones de contexto y MQMD-Descriptor de mensaje para obtener descripciones de los campos de descriptor de mensaje relacionados con el contexto.

Implementación de control de accesos en salidas de seguridad

Puede implementar control de accesos en una salida de seguridad utilizando el campo MCAUserIdentifier o el gestor de autorizaciones sobre objetos.

MCAUserIdentifier

Cada instancia de un canal actual tiene una estructura de definición de canal, MQCD, asociada. Los valores iniciales de los campos de MQCD se determinan mediante la definición de canal que crea un administrador de IBM MQ. En particular, el valor inicial de uno de los campos, *MCAUserIdentifier* , se determina mediante el valor del parámetro MCAUSER del mandato DEFINE CHANNEL o mediante el equivalente a MCAUSER si la definición de canal se crea de otra forma.

La estructura MQCD se pasa a un programa de salida de canal al que llama un MCA. Cuando un MCA llama a una salida de seguridad, la salida de seguridad puede cambiar el valor de *MCAUserIdentifier*, sustituyendo el valor especificado en la definición de canal.

Multi En Multiplatforms, a menos que el valor de *MCAUserIdentifier* esté en blanco, el gestor de colas utiliza el valor de *MCAUserIdentifier* como ID de usuario para las comprobaciones de autorización cuando un MCA intenta acceder a los recursos del gestor de colas después de que se haya conectado al gestor de colas. Si el valor de *MCAUserIdentifier* está en blanco, el gestor de colas utiliza en su lugar el ID de usuario predeterminado del MCA. Esto es aplicable a los canales RCVR, RQSTR, CLUSRCVR y SVRCONN. Para los MCA emisores, el ID de usuario predeterminado se utiliza siempre para las comprobaciones de autorización, incluso si el valor de *MCAUserIdentifier* no está en blanco.

z/OS En z/OS, el gestor de colas puede utilizar el valor de *MCAUserIdentifier* para comprobaciones de autorización, siempre y cuando no esté en blanco. Para los MCA receptores y los MCA de conexión con servidor, el hecho de que el gestor de colas utilice el valor de *MCAUserIdentifier* para comprobaciones de autoridad depende de:

- El valor del parámetro PUTAUT en la definición de canal
- El perfil RACF utilizado para las comprobaciones
- El nivel de acceso del ID de usuario del espacio de direcciones del iniciador de canal ante el perfil RESLEVEL

Para los MCA emisores, depende de:

- Si el MCA emisor efectúa la llamada o envía la respuesta
- El nivel de acceso del ID de usuario del espacio de direcciones del iniciador de canal ante el perfil RESLEVEL

El ID de usuario que una salida de seguridad almacena en *MCAUserIdentifier* se puede adquirir de varias formas. A continuación, se detallan algunos ejemplos:

- Suponiendo que no hay ninguna salida de seguridad en el extremo del cliente de un canal MQI, un ID de usuario asociado con la aplicación cliente IBM MQ fluye desde el MCA de conexión del cliente al MCA de conexión del servidor cuando la aplicación cliente emite una llamada MQCONN.El MCA de conexión del servidor almacena su ID de usuario en el campo *RemoteUserIdentifier* de la estructura de definición de canal, MQCD. Si el valor de *MCAUserIdentifier* está en blanco en este momento, el MCA almacena el mismo ID de usuario en *MCAUserIdentifier*. Si el MCA no almacena el ID de usuario en *MCAUserIdentifier*, una salida de seguridad puede hacerlo posteriormente, estableciendo *MCAUserIdentifier* en el valor de *RemoteUserIdentifier*.

  Si el ID de usuario que fluye del sistema cliente está entrando en un nuevo dominio de seguridad y no es válido en el sistema servidor, la salida de seguridad puede sustituir el ID de usuario por uno que sea válido y almacenar el ID de usuario sustituido en *MCAUserIdentifier*.
- La salida de seguridad del asociado puede enviar el ID de usuario en un mensaje de seguridad.

  En un canal de mensaje, una salida de seguridad a la que ha llamado el MCA emisor puede enviar el ID de usuario bajo el cual se ejecuta el MCA emisor. Luego, una salida de seguridad a la que ha llamado el MCA receptor puede almacenar el ID de usuario en *MCAUserIdentifier* . De forma similar, en un canal MQI, una salida de seguridad en el extremo del cliente del canal puede enviar el ID de usuario asociado con la aplicación IBM MQ MQI client. Luego una salida de seguridad en el extremo del servidor del canal puede almacenar el ID de usuario en *MCAUserIdentifier*. Como en el ejemplo anterior, si el ID de usuario no es válido en el sistema de destino, la salida de seguridad puede sustituir el ID de usuario por uno que sea válido y almacenar el ID de usuario sustituido en *MCAUserIdentifier*.

Si se recibe un certificado digital como parte del servicio de identificación y autenticación, una salida de seguridad puede correlacionar el Nombre distinguido del certificado con un ID de usuario que sea válido en el sistema de destino. Puede almacenar el ID de usuario en *MCAUserIdentifier*.

- Si TLS se utiliza en el canal, el nombre distinguido del asociado (DN) se pasa a la salida del campo SSLPeerNamePtr de MQCD y el DN de emisor del certificado se pasa a la salida del campo SSLRemCertIssNamePtr de MQCXP.

Para obtener más información sobre el campo *MCAUserIdentifier*, la estructura de definición de canal, MQCD, y la estructura del parámetro de salida de canal, MQCXP, consulte Llamadas de salida de canal y estructuras de datos. Para obtener más información sobre el ID de usuario que fluye desde un sistema cliente en un canal MQI, consulte Control de accesos.

Nota: Las aplicaciones de salida de seguridad creadas antes del release de IBM WebSphere MQ 7.1 pueden requerir actualización. Para obtener más información, consulte Programas de salida de seguridad de canal.

Autenticación de usuarios del gestor de autorizaciones sobre objetos de IBM MQ

En las conexiones de cliente MQI de IBM MQ MQI client, las salidas de seguridad se pueden utilizar para modificar o crear la estructura MQCSP utilizada en una autenticación de usuario del gestor de autorizaciones sobre objetos (OAM). Esto se describe en la sección Programas de salida de canal para canales de mensajería

Implementación de control de accesos en salidas de mensajes

Es posible que tenga que utilizar una salida de mensajes para sustituir un ID de usuario por otro.

Considere una aplicación cliente que envía un mensaje a una aplicación de servidor. La aplicación de servidor puede extraer el ID de usuario del campo *UserIdentifier* del descriptor de mensaje y, siempre y cuando tenga autorización de usuario alternativo, solicitar al gestor de colas que utilice este ID de usuario para comprobaciones de autorización cuando acceda a recursos de IBM MQ en nombre del cliente.

Si el parámetro PUTAUT se establece en CTX (o ALTMCA en z/OS) en la definición de canal, el ID de usuario del campo *UserIdentifier* de cada mensaje de entrada se utiliza para las comprobaciones de autorización cuando el MCA abre la cola de destino.

En determinadas circunstancias, cuando se genera un mensaje de informe, este se coloca utilizando la autoridad del ID de usuario del campo *UserIdentifier* del mensaje que ha originado el informe. En particular, los informes de confirmación de entrega (COD) y los informes de caducidad siempre se colocan con esta autoridad.

Debido a estas situaciones, es posible que sea necesario sustituir un ID de usuario por otro en el campo *UserIdentifier* cuando un mensaje entra en un nuevo dominio de seguridad. Esto se puede hacer mediante una salida de mensajes en el extremo receptor del canal. Como alternativa, puede asegurarse de que el ID de usuario del campo *UserIdentifier* de un mensaje de entrada está definido en el nuevo dominio de seguridad.

Si un mensaje de entrada contiene un certificado digital correspondiente al usuario de la aplicación que ha enviado el mensaje, una salida de mensajes puede validar el certificado y correlacionar el Nombre distinguido del certificado con un ID de usuario que sea válido en el sistema receptor. Luego puede establecer el campo *UserIdentifier* del descriptor de mensajes en este ID de usuario.

Si es necesario que una salida de mensajes cambie el valor del campo *UserIdentifier* en un mensaje de entrada, es posible que la salida de mensajes tenga que autenticar el emisor del mensaje al mismo tiempo. Para obtener más detalles, consulte “Correlación de identidad en salidas de mensajes” en la página 330.

Implementación de control de accesos en la salida de API y la salida cruzada de API

Una salida de API o una salida cruzada de API puede proporcionar controles de accesos para complementar los que proporciona IBM MQ. En concreto, la salida puede proporcionar control de accesos a nivel de mensaje. La salida puede garantizar que una aplicación transfiera a una cola, u obtenga de una cola, sólo aquellos mensajes que cumplen ciertos criterios.

Tenga en cuenta los ejemplos siguientes:

- Un mensaje contiene información sobre un pedido. Cuando una aplicación intenta transferir un mensaje a una cola, una salida de API o salida cruzada de API puede comprobar si el valor total del pedido es inferior a un límite establecido previamente.
- Llegan mensajes a una cola de destino desde gestores de colas remotos. Cuando una aplicación intenta obtener un mensaje de la cola, una salida de API o salida cruzada de API puede comprobar si el emisor del mensaje tiene autorización para enviar un mensaje a la cola.

Multi Seguridad de colas de transmisión

La característica de colas de modalidad continua permite a un administrador configurar una cola local (o modelo) con una cola secundaria, donde se colocan los mensajes duplicados, siempre que se coloca un mensaje en la cola original. Hay dos aspectos a tener en cuenta con respecto a las autorizaciones de streaming de cola.

Autorización para configurar una cola para transmitir mensajes duplicados

Si desea habilitar la modalidad continua de mensajes duplicados de una cola a una cola secundaria, debe tener permiso para hacerlo. El permiso para configurar el atributo **STREAMQ** de una cola requiere que tenga las autorizaciones siguientes:

1. Autorización CHG de la cola para la que están alterando el atributo **STREAMQ**
2. Autorización CHG de la cola en la que desea que se transfieran los mensajes de duplicación

La combinación de estas dos comprobaciones de autorización en el momento de la configuración garantiza que un usuario, que sólo tiene autorización CHG en la cola original, no puede hacer que los mensajes se transfieran a otra cola en la que no tienen permisos.

Autorización para abrir la cola o colas y transferir mensajes

Cuando una aplicación abre una cola que se ha configurado con una cola secundaria, a través de su atributo **STREAMQ** , se realiza una comprobación de autorización de que el usuario de la aplicación tiene autorización PUT sobre la cola original.

Nota: No se realiza ninguna comprobación de autorización adicional para el usuario de la aplicación en la cola secundaria, que es similar al modelo de autorización utilizado para las colas alias.

Las aplicaciones que consumen mensajes de la cola original o secundaria requieren autorización GET o BROWSE, sólo en la cola de la que están consumiendo.

No se realizan comprobaciones de autorización adicionales en el momento de la puesta o de la obtención.

Ejemplo

El ejemplo siguiente muestra las autorizaciones correctas que se establecen para permitir al usuario `admin` configurar una cola original, INQUIRIES.QUEUE, para transmitir sus mensajes duplicados a la cola local ANALYTICS.QUEUE, pero impidiendo que `admin` duplique los mensajes en 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)
```

A continuación, el usuario admin puede emitir el mandato siguiente:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

pero si el mismo usuario emite el mandato siguiente:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

para configurar INQUIRIES.QUEUE para colocar mensajes duplicados en PURCHASES.QUEUE, reciben el siguiente error:

```
AMQ8135E No autorizado
```

Con INQUIRIES.QUEUE configurado para duplicar mensajes en ANALYTICS.QUEUE, los siguientes registros de autorización se utilizan para permitir que una aplicación que se ejecuta como usuario appuser transfiera mensajes a INQUIRIES.QUEUEy mensajes duplicados en ANALYTICS.QUEUE:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

Nota: appuser no requiere un registro de autorización en ANALYTICS.QUEUE. El gestor de colas coloca los mensajes duplicados en la cola.

Conceptos relacionados

Colas de modalidad continua

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 Autorización LDAP

Puede utilizar la autorización LDAP para eliminar la necesidad de un ID de usuario local.

Disponibilidad de la autorización LDAP en plataformas soportadas

La autorización LDAP está disponible en Multiplatforms:

Atención:

Desde la disponibilidad general de IBM MQ 9.0, esta funcionalidad está disponible en todos los gestores de colas, ya sean nuevos o migrados de un release anterior.

Visión general de la autorización LDAP

Con la autorización LDAP, los mandatos que manejan la configuración de autorización, como por ejemplo **setmqaut** y **DISPLAY AUTHREC**, pueden procesar nombres distinguidos. Anteriormente, los usuarios se autenticaban comparando sus credenciales con el máximo de caracteres disponibles que existen para los usuarios y grupos en el sistema operativo local.

Atención: Si ha ejecutado el mandato **DEFINE AUTHINFO**, debe reiniciar el gestor de colas. Si no se reinicia el gestor de colas, el mandato setmqaut no devuelve el resultado correcto.

Si un usuario proporciona un ID de usuario en lugar de un nombre distinguido, el ID de usuario se procesa. Por ejemplo, cuando haya un mensaje entrante en un canal con PUTAUT(CTX), los caracteres en el ID de usuario se correlacionan con un nombre distinguido LDAP y se realizan las comprobaciones de autorización correspondientes.

Otros mandatos como **DISPLAY CONN**, siguen trabajando con el ID de usuario y muestran el valor real para el ID de usuario, aunque este ID de usuario pueda no existir realmente en el sistema operativo local.

Linux AIX Cuando la autorización LDAP se aplica, el gestor de colas siempre utiliza el modelo de usuario de seguridad en plataformas AIX and Linux, independientemente del atributo **SecurityPolicy** en el archivo qm.ini. Por lo tanto, el establecimiento de permisos para un usuario

individual sólo afecta a ese usuario, y no a nadie más que pertenezca a alguno de los grupos de ese usuario.

Al igual que con el modelo del sistema operativo, un usuario aún tiene la autorización combinada que se ha asignado tanto a las personas como a los grupos (si hay alguno) a los que pertenece el usuario.

Por ejemplo, suponga que se han definido los siguientes registros en un repositorio LDAP.

- En la clase **inetOrgPerson**:

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- En la clase **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"
```

A efectos de autenticación, se debe haber definido un gestor de colas que utiliza este servidor LDAP de modo que su valor **CONNAUTH** apunte a un objeto **AUTHINFO** del tipo IDPWLDAP, y cuyos atributos de resolución de nombres relevantes probablemente se establezcan de la forma siguiente:

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

Con esta configuración de autenticación, una aplicación puede completar el campo CSPUserID, que se utiliza dentro de la llamada MQCNO, con uno de los siguientes conjuntos de valores:

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

o

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

En cualquiera de los casos, el sistema puede utilizar los valores proporcionados para autenticar el contexto de sistema operativo de "jodoe".

Multi Configuración de autorizaciones

Cómo utilizar el nombre abreviado o **USRFIELD** para definir autorizaciones.

El método de trabajar con varios formatos, que se describe en “Autorización LDAP” en la página 421, continúa en los mandatos de autorización, con una extensión adicional que el shortname o el USRFIELD se pueden utilizar de forma no adornada.

La serie de caracteres especifica un atributo concreto en el registro LDAP cuando se especifican usuarios (principales) para la autorización.

Importante: La serie de caracteres no debe contener el carácter =, porque este carácter no se puede utilizar en un ID de usuario del sistema operativo.

Si pasa un nombre principal al OAM para la autorización que es potencialmente un shortname, la serie de caracteres debe caber en 12 caracteres. El algoritmo de correlación primero intenta resolverlo en un DN utilizando el atributo SHORTUSR en su consulta LDAP.

Si esto falla con un error UNKNOWN_ENTITY, o si la serie dada no puede ser posiblemente un shortname, se realiza un intento adicional utilizando el atributo USRFIELD para construir la consulta LDAP.

Atención: Si se ha ejecutado el mandato DEFINE AUTHINFO, hay que reiniciar el gestor de colas. Si no se reinicia el gestor de colas, el mandato setmqaut no devuelve el resultado correcto.

A la hora de procesar las autorizaciones de usuario, los siguientes valores del mandato setmqaut son equivalentes.

Tabla 75. Valores de autorización de usuario

Mandato	Nota
`setmqaut -m QM -t qmgr -p jodoe +connect`	Es un nombre plano no calificado, resuelto a través de SHORTUSR.
`setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect`	También un nombre plano no calificado, que se resuelve a través de USRFIELD en la misma entidad,
`setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect`	Utilizando un atributo especificado.
`setmqaut -m QM -t qmgr -p "phone=1234567" +connect`	Utilizando otro atributo especificado que no tiene que ser ninguno de los configurados en el objeto AUTHINFO.

Puede utilizar el mandato SET AUTHREC MQSC como alternativa al mandato **setmqaut**:

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

o el mandato PCF Establecer registro de autorización (MQCMD_SET_AUTH_REC) con el elemento MQCACF_PRINCIPAL_ENTITY_NAMES que contiene la serie:

```
 "cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

Al procesar grupos, no hay ambigüedad sobre el proceso de `shortname` , ya que no hay ningún requisito para ajustar cualquier forma de un nombre de grupo en 12 caracteres. Por lo tanto, no existe ningún equivalente del atributo SHORTUSR para grupos.

Esto significa que los ejemplos de sintaxis descritos en Tabla 76 en la página 423 son válidos suponiendo que ha se configurado el objeto AUTHINFO con los atributos ampliados y se ha establecido en:

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

Tabla 76. Valores de autorización de grupo

Mandato	Nota
`setmqaut -m QM -t qmgr -g ApplicationGroupA +connect`	Utilizando GRPFIELD para resolver
`setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect`	Especificando un solo atributo
`setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect`	Utilizando el nombre distinguido completo

Puede utilizar el mandato MQSC SET AUTHREC como alternativa al mandato **setmqaut** anterior:

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

o el mandato PCF Establecer registro de autorización (MQCMD_SET_AUTH_REC) con el elemento MQCACF_GROUP_ENTITY_NAMES que contiene la serie:

```
 "ApplicationGroupA"
```

Importante:

Sea cual sea el formato que se utiliza para hacer referencia a un nombre, ya sea para usuario o grupo, debe ser posible obtener un DN exclusivo.

Así, por ejemplo, no debe tener dos registros distintos que sean "`shortu=jodoe`".

Si no puede determinarse un solo nombre distinguido exclusivo, el OAM devuelve MQRC_UNKNOWN_ENTITY.

Multi Visualización de autorizaciones

Hay distintos métodos de visualizar autorizaciones de usuarios o grupos.

Mandato dspmqaut

El método más simple para visualizar las autorizaciones que está disponible para un usuario o grupo es utilizar el mandato dspmqaut.

Puede utilizar una consulta en cualquiera de las variaciones de sintaxis para identificar un usuario o un grupo. Tenga en cuenta que la salida del mandato repite la identidad en el formato especificado en la línea de mandatos. La salida no informa sobre el nombre distinguido completo resuelto.

Por ejemplo:

```
dspmqaut -m QM -t qmgr -p johndoe
Entity johndoe has the following authorizations for object QM:
    connect
```

o

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

Mandatos dmpmqaut y dmpmqcfg

El mandato dmpmqaut y sus mandatos MQSC o PCF equivalentes, puede especificar el principal o el grupo en cualquiera de los formatos soportados, como las tablas de **setmqaut** descritas en “Configuración de autorizaciones” en la página 422. Sin embargo, a diferencia de **dspmqaut**, el mandato **dmpmqaut** siempre notifica el nombre distinguido 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
```

Del mismo modo, el mandato dmpmqcfg, que no tiene ningún filtro en los registros seleccionados, siempre muestra el nombre distinguido completo en un formato que se pueden reproducir más adelante.

```
dmpmqcfg -m QM -x authrec
-----------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

Multi

Otras consideraciones al utilizar la autorización LDAP

Breve descripción de los cambios en la interfaz de cola de mensajes (Message Queue Interface, MQI) y otros mandatos MQSC y PCF necesarios para saber cuándo se usa una autorización LDAP desde IBM MQ 9.0.0.

ADOPTCTX

No hay ningún requisito para que las aplicaciones proporcionen información de autenticación, o para que el atributo ADOPTCTX se establezca en YES.

Si una aplicación no se autentica explícitamente, o si **ADOPTCTX** se establece en NO para el objeto CONNAUTH activo, el contexto de identidad asociado con la aplicación se toma del ID de usuario del sistema operativo.

Cuando es necesario aplicar autorizaciones, dicho contexto se correlaciona con una entidad LDAP utilizando las mismas reglas que para los mandatos setmqaut.

Parámetros de entrada para llamadas MQI

MQOPEN, MQPUT1 y MQSUB tienen estructuras que permiten especificar un ID de usuario alternativo.

Si se utilizan estos campos, el ID de usuario de 12 caracteres se correlaciona con un DN utilizando las mismas reglas que en los mandatos **setmqaut**, **dmpmqaut**y **dspmqaut** .

MQPUT y MQPUT1 también permiten programas con la autorización adecuada establecer el campo de MQMD UserIdentifier. El valor de este campo no se supervisará durante el proceso PUT y puede establecerse en cualquier valor.

Sin embargo, como es habitual, el valor **UserIdentifier** se puede utilizar para la autorización en fases posteriores del proceso de mensajes, por ejemplo cuando PUTAUT(CTX) se define en un canal receptor.

En dicho momento, se comprobará la autorización del identificador del receptor utilizando la configuración de dicho gestor de colas receptor, que puede ser basado en sistema operativo o LDAP.

Parámetros de salida para llamadas MQI

Siempre que se proporciona un ID de usuario a un programa en una estructura MQI, es la versión de nombre abreviado de 12 caracteres asociada a la conexión.

Por ejemplo, el valor **MQAXC.UserId** para las salidas de API es el nombre abreviado devuelto de la correlación LDAP.

Otros mandatos MQSC y PCF administrativos

Los mandatos que muestran información de usuario en estado de objeto como DISPLAY CONN USERID devuelven el nombre abreviado de 12 caracteres asociado con el contexto. El nombre distinguido completo no se muestra.

Los mandatos que permiten la aserción de identidades, como las reglas de correlación CHLAUTH o los valores MCAUSER para canales, pueden tomar valores hasta llegar a la longitud máxima definida para estos atributos (actualmente 64 caracteres).

No hay ningún cambio en la sintaxis. Cuando es necesaria la autorización para esa identidad, ésta se correlaciona internamente con un nombre distinguido utilizando las mismas reglas que para los mandatos **setmqaut**, **dmpmqaut** y **dspmqaut**.

Esto significa que el valor MCAUSER en una definición de canal puede no visualizarse como la misma serie que DISPLAY CHSTATUS aunque hacen referencia a la misma identidad.

Por ejemplo:

```
DEFINE CHL(SV1) CHLTYPE(SVRCONN) MCAUSER('cn=JohnDoe')
DEFINE CHL(SV2) CHLTYPE(SVRCONN) MCAUSER('jodoe')
DEFINE CHL(SV3) CHLTYPE(SVRCONN) MCAUSER('JohnDoe1@yourcompany.com')
```

A continuación DISPLAY CHSTATUS(*) ALL mostrará el valor SHORTUSR, *MCAUSER(jodoe)* para todas las conexiones.

Multi Conmutación entre modelos de autorización del sistema operativo y LDAP

Cómo se puede conmutar entre distintos métodos de autorización en distintas plataformas.

El atributo CONNAUTH del gestor de colas apunta a un objeto AUTHINFO. Cuando el objeto es del tipo IDPWLDAP, se utiliza un repositorio LDAP para la autenticación.

Ahora puede aplicar un método de autorización al mismo objeto, lo que le permite continuar con la autorización basada en sistema operativo, o trabajar con autorización LDAP

IBM i, AIX and Linux

Linux IBM i AIX

El gestor de colas puede cambiarse en cualquier momento entre los modelos de sistema operativo y LDAP. Puede cambiar la configuración y hacer que dicha configuración sea la activa mediante el mandato REFRESH SECURITY TYPE (CONNAUTH).

Por ejemplo, si este objeto ya se ha configurado con la información de conexión para la autenticación:

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

Windows

Windows

Si un cambio de configuración de autorización implica la conmutación entre modelos de sistema operativo y LDAP, el gestor de colas debe reiniciarse para que el cambio entre en vigor. de lo contrario, puede activar el cambio mediante el mandato REFRESH SECURITY TYPE (CONNAUTH).

Reglas de proceso

Cuando se cambia de la autorización de sistema operativo a la autorización LDAP, todas las reglas de autorización de sistema operativo existentes que se han establecido pasan a estar inactivas e invisibles.

Los mandatos tales como **dmpmqaut** no visualizan estas reglas de sistema operativo. De forma similar, cuando se vuelve a pasar de LDAP a SO, todas las autorizaciones LDAP definidas pasan a estar inactivas e invisibles, y se restauran las reglas de sistema operativo originales.

Si desea hacer copia de seguridad de las definiciones de un gestor de colas por alguna razón, con el mandato **dmpmqcfg**, esta copia de seguridad solamente incluirá las reglas definidas para el método de autorización en efecto en el momento de realizar la copia de seguridad.

Multi Administración LDAP

Una visión general de cómo cada plataforma administra LDAP.

Cuando se utiliza la autorización LDAP, el miembro del grupo `mqm` (o equivalente) en el sistema operativo no es muy importante. Ser miembro de dicho grupo solamente controla si se pueden procesar determinados mandatos de línea de mandatos.

En concreto, debe ser miembro de dicho grupo para emitir mandatos strmqm y endmqm.

Una vez que el gestor de colas está en ejecución, ahora hay límites en la cuenta con todos los privilegios. Aparte del ID de usuario de la persona que emite el mandato **strmqm**, los otros usuarios que pertenecen al grupo `mqm` (o equivalente) del sistema operativo no tienen privilegios especiales.

Las autorizaciones de otros usuarios se basan en a qué grupos LDAP pertenecen. Un uso no calificado del nombre de grupo `mqm` en mandatos tales como **setmqaut** no está permitido para correlacionar con ningún grupo LDAP.

AIX and Linux

Linux AIX

Una vez que el gestor de colas está en ejecución, la única cuenta que tiene todos los privilegios automáticamente es el usuario real que ha iniciado el gestor de colas.

El ID `mqm` sigue existiendo y se utiliza como el propietario de recursos del sistema operativo, tales como archivos, porque `mqm` es el ID efectivo bajo el que se ejecuta el gestor de colas. Sin embargo, el usuario `mqm` no podrá realizar automáticamente tareas administrativas controladas por el OAM.

Windows

Windows

En Windows, las cuentas con privilegios completos automáticos son la del usuario del sistema operativo que ha iniciado el gestor de colas, y también la del usuario que ejecuta los procesos principales del gestor de colas, como por ejemplo MUSR_MQADMIN si el gestor de colas se ha iniciado como un servicio de Windows.

Cuando se ejecuta en modalidad de autorización LDAP, Windows se comporta de forma muy parecida a las plataformas AIX and Linux. Se ocupa de nombres abreviados de 12 caracteres y nombres distinguidos completos.

IBM i

IBM i

En IBM i, las cuentas que tienen automáticamente privilegios son las que inician el gestor de colas y el ID de QMQM.

Son necesarios los dos ID porque el ID de usuario que inicia el gestor de colas solamente es necesario para iniciar el sistema. Una vez que está en ejecución, los procesos del gestor de colas solo tienen autorización QMQM.

Script de ejemplo para proporcionar privilegios MQADMIN

Linux AIX

Como es útil tener un grupo que sea capaz de realizar la administración completa en un gestor de colas, se entrega un script de ejemplo en las plataformas AIX and Linux como:

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

Este ejemplo tiene dos parámetros:

- Un nombre de gestor de colas
- Un nombre de grupo LDAP

El ejemplo procesa mandatos setmqaut, y otorga autorización total para todos los objetos. Se trata del mismo script generado por el asistente de OAM de IBM MQ Explorer para funciones administrativas. Por ejemplo, el código empieza:

```
setmqaut -t q -m qmgr -n "**" +alladm -g
      groupname
```

Confidencialidad de mensajes

El cifrado de mensajes garantiza que el contenido de los mensajes permanece confidencial. Existen diversos métodos de cifrado de mensajes en IBM MQ, en función de sus necesidades.

Si necesita protección de datos de extremo a extremo a nivel de aplicación para la infraestructura de mensajería de punto a punto, puede utilizar Advanced Message Security para cifrar los mensajes o escribir su propia salida de API o salida cruzada de API.

La solución más segura es proporcionar cifrado de extremo a extremo, cifrando un mensaje desde el punto en el que lo coloca una aplicación, hasta el punto en el que lo recibe la aplicación consumidora. Esto se puede realizar utilizando “Planificación de Advanced Message Security” en la página 115 (AMS), o escribiendo su propia salida de API o salida cruzada de API; consulte “Implementación de confidencialidad en programas de salida de usuario” en la página 476 para obtener más información.

Si necesita cifrar mensajes sólo mientras se transportan a través de una red, puede utilizar TLS; consulte “Protocolos de seguridad TLS en IBM MQ” en la página 25 para obtener más información, o puede escribir su propia salida de seguridad, salida de mensajes o programas de salida de envío y recepción para realizar el cifrado.

z/OS Si necesita cifrar mensajes en reposo en un gestor de colas, puede utilizar el cifrado de conjuntos de datos de z/OS en ese gestor de colas; consulte “Confidentiality for data at rest on IBM MQ for z/OS with data set encryption” en la página 478 para obtener más información.

Tareas relacionadas

Conexión de dos gestores de colas utilizando TLS

Conexión de un cliente a un gestor de colas de forma segura

Habilitación de CipherSpecs

Habilite una CipherSpec utilizando el parámetro **SSLCIPH** en el mandato **DEFINE CHANNEL** o **ALTER CHANNEL** MQSC.

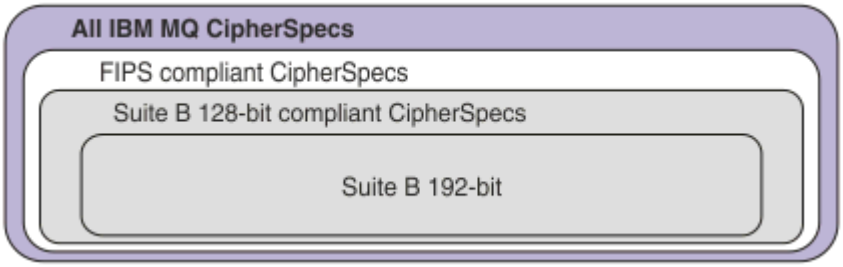
Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC) . El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el certificado deIBM Crypto for C (ICC) y tener en cuenta cualquier consejo proporcionado por NIST. Un módulo FIPS 140-3 de sustitución está actualmente en curso y su estado se puede ver buscándolo en los módulos CMVP de NIST en la lista de procesos.

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los módulos CMVP de NIST en la lista de procesos.

Algunas de las CipherSpecs que puede utilizar con IBM MQ son compatibles con FIPS. Algunas de las CipherSpecs compatibles con FIPS también son compatibles con Suite B aunque otras, como TLS_RSA_WITH_AES_256_CBC_SHA, no lo son.

Todas las CipherSpecs compatibles con Suite B también son compatibles con FIPS.
Todas las CipherSpecs compatibles con Suite B se clasifican en dos grupos:
128 bits (por ejemplo, ECDHE_ECDSA_AES_128_GCM_SHA256 y 192 bits (por ejemplo, ECDHE_ECDSA_AES_256_GCM_SHA384),

El siguiente diagrama ilustra la relación entre estos subconjuntos:



El producto da soporte al protocolo de seguridad TLS 1.3 en todas las plataformas.

Las CipherSpecs que puede utilizar para cada una de estas plataformas se listan en Tabla 77 en la página 430. Para obtener información sobre la utilización de estas CipherSpecs, consulte “Utilización de TLS 1.3 en IBM MQ” en la página 433 y “IBM MQ MQI client y TLS 1.3” en la página 433.

Para facilitar la configuración y la migración futura, IBM MQ también proporciona un conjunto de alias CipherSpecs. La migración de configuraciones de seguridad existentes para utilizar un CipherSpec de alias significa que puede adaptarse a condiciones de adición y desuso del cifrado sin necesidad de realizar más cambios de configuración invasivos en el futuro. Estos alias CipherSpecs se listan en la sección CipherSpecs de alias en Tabla 77 en la página 430. Para obtener más información sobre la migración para utilizar un alias CipherSpec, consulte Migración de configuraciones de seguridad existentes para utilizar un alias CipherSpec.

Puede configurar las CipherSpecs predeterminadas tal como se describe en “Valores CipherSpec predeterminados habilitados en IBM MQ” en la página 434. También puede proporcionar un conjunto alternativo de CipherSpecs que están habilitadas para su uso con canales en:

- Multi IBM MQ for Multiplatforms, tal como se describe en “Proporcionar una lista personalizada de CipherSpecs ordenadas y habilitadas en IBM MQ for Multiplatforms” en la página 442.
- z/OS IBM MQ for z/OS, tal como se describe en “Proporcionar una lista personalizada de CipherSpecs ordenadas y habilitadas en IBM MQ for z/OS” en la página 443.

Las CipherSpecs en desuso que puede volver a habilitar para utilizarlas con IBM MQ si es necesario se listan en “CipherSpecs en desuso” en la página 444.

CipherSpecs que se pueden utilizar con el soporte TLS de IBM MQ

Las CipherSpecs que puede utilizar con el gestor de colas IBM MQ se listan automáticamente en la tabla siguiente. Cuando solicite un certificado personal, especifique un tamaño de clave para el par de claves pública y privada. El tamaño de clave que se utiliza durante el reconocimiento TLS es el tamaño almacenado en el certificado a menos que esté determinado por la CipherSpec, tal como está indicado en la tabla.

Tabla 77. CipherSpecs que puede utilizar con el soporte TLS de IBM MQ

Soporte de plataforma "1" en la página 432	Nombre de CipherSpec	Código hexadecimal	Protocolo utilizado	Algoritmo MAC	Algoritmo de cifrado (bits de cifrado)	FIPS "2" en la página 432	Suite B
CipherSpecs de alias							
Todo	ANY_TLS13_OR_HIGHER "3" en la página 432 "4" en la página 432	No disponible	Negociado	Negociado	Negociado	Negociado	Negociado
Todo	ANY_TLS13 "4" en la página 432 "5" en la página 432	No disponible	TLS 1.3	Negociado	Negociado	Negociado	Negociado
Todo	ANY_TLS12_OR_HIGHER "4" en la página 432 "6" en la página 432	No disponible	Negociado	Negociado	Negociado	Negociado	Negociado
Todo	ANY_TLS12 "7" en la página 432	No disponible	TLS 1.2	Negociado	Negociado	Negociado	Negociado
Todo	ANY "8" en la página 432	No disponible	Negociado	Negociado	Negociado	Negociado	Negociado
CipherSpecs for TLS 1.3							
Todo	TLS_AES_128_GCM_SHA256	1301	TLS 1.3	GCM	AES-128 con GCM (128)	Sí	No
Todo	TLS_AES_256_GCM_SHA384	1302	TLS 1.3	GCM	AES-256 con GCM (256)	Sí	No
Todo	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 con CTR (128)	Sí	No
ALW	TLS_AES_128_CCM_8_SHA256 "10" en la página 432	1305	TLS 1.3	CBC-MAC	AES-128 con CTR (128)	Sí	No
CipherSpecs para TLS 1.2							
Todo	TLS_RSA_WITH_AES_128_CBC_SHA256 "9" en la página 432	003C	TLS 1.2	SHA-256	AES (128)	Sí	No
Todo	TLS_RSA_WITH_AES_256_CBC_SHA256 "9" en la página 432 "11" en la página 432	003D	TLS 1.2	SHA-256	AES (256)	Sí	No
Todo	TLS_RSA_WITH_AES_128_GCM_SHA256 "9" en la página 432 "12" en la página 432	009C	TLS 1.2	SHA-256 y AEAD GCM	AES (128)	Sí	No

Tabla 77. CipherSpecs que puede utilizar con el soporte TLS de IBM MQ (continuación)

Soporte de plataforma "1" en la página 432	Nombre de CipherSpec	Código hexadecimal	Protocolo utilizado	Algoritmo MAC	Algoritmo de cifrado (bits de cifrado)	FIPS "2" en la página 432	Suite B
Todo	TLS_RSA_WITH_AES_256_GCM_SHA384 "9" en la página 432 "11" en la página 432 "12" en la página 432	009D	TLS 1.2	SHA-384 y AEAD GCM	AES (256)	Sí	No
Todo	ECDHE_ECDSA_AES_128_CBC_SHA256 "9" en la página 432	C023	TLS 1.2	SHA-256	AES (128)	Sí	No
Todo	ECDHE_ECDSA_AES_256_CBC_SHA384 "9" en la página 432 "11" en la página 432	C024	TLS 1.2	SHA-384	AES (256)	Sí	No
Todo	ECDHE_RSA_AES_128_CBC_SHA256 "9" en la página 432	C027	TLS 1.2	SHA-256	AES (128)	Sí	No
Todo	ECDHE_RSA_AES_256_CBC_SHA384 "9" en la página 432 "11" en la página 432	C028	TLS 1.2	SHA-384	AES (256)	Sí	No
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 "11" en la página 432 "12" en la página 432	C02B	TLS 1.2	SHA-256 y AEAD GCM	AES (SHA384)	Sí	128 bits
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384 "11" en la página 432 "12" en la página 432	C02C	TLS 1.2	SHA-384 y AEAD GCM	AES (SHA384)	Sí	192 bits
Todo	ECDHE_RSA_AES_128_GCM_SHA256 "12" en la página 432	C02F	TLS 1.2	SHA-256 y AEAD GCM	AES (128)	Sí	No
Todo	ECDHE_RSA_AES_256_GCM_SHA384 "11" en la página 432 "12" en la página 432	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	Sí	No

Tabla 77. CipherSpecs que puede utilizar con el soporte TLS de IBM MQ (continuación)

Soporte de plataforma "1" en la página 432	Nombre de CipherSpec	Código hexadecimal	Protocolo utilizado	Algoritmo MAC	Algoritmo de cifrado (bits de cifrado)	FIPS "2" en la página 432	Suite B

Notas:

1. Para obtener una lista de las plataformas cubiertas por cada icono de plataforma, consulte Iconos utilizados en la documentación del producto.
2. Especifica si la CipherSpec tiene el certificado FIPS en una plataforma certificada con FIPS. Consulte Federal Information Processing Standards (FIPS - Estándares federales de procesamiento de información) para obtener una explicación de FIPS.
3. ALW El alias ANY_TLS13_OR_HIGHER de CipherSpec negocia el mayor nivel de seguridad que el extremo remoto permitirá, pero solo se conectará utilizando un protocolo TLS 1.3 o superior.
4. IBM i Para utilizar TLS 1.3 en IBM i, la versión del sistema operativo subyacente debe soportar TLS 1.3. Consulte Soporte del sistema TLS para TLSv1.3 para obtener más información.
5. ALW El alias ANY_TLS13 de CipherSpec representa un subconjunto de CipherSpecs aceptables que utilizan el protocolo TLS 1.3, como se enumeran en esta tabla para cada plataforma.
6. ALW El alias ANY_TLS12_OR_HIGHER de CipherSpec negocia el mayor nivel de seguridad que el extremo remoto permitirá, pero solo se conectará utilizando un protocolo TLS 1.2 o superior.
7. La ANY_TLS12 CipherSpec representa un subconjunto de CipherSpecs aceptables que utilizan el protocolo TLS 1.2, como aparecen listadas en esta tabla para cada plataforma.
8. ALW El alias ANY de CipherSpec negocia el mayor nivel de seguridad que el extremo remoto permitirá.
9. IBM i Estas CipherSpecs no están habilitadas en los sistemas IBM i 7.4 que tienen el valor del sistema QSSLCSLCTL establecido en *OPSSYS.
10. ALW Estas CipherSpecs utilizan un valor de comprobación de integridad (ICV) de 8 octetos en lugar de un ICV de 16 octetos.
11. Esta CipherSpec no se puede utilizar para garantizar una conexión desde IBM MQ Explorer a un gestor de colas amenos que se apliquen los archivos de políticas no restringidas apropiados al JRE utilizado por Explorer.
12. ALW Siguiendo una recomendación de GSKit, TLS 1.2 GCM CipherSpecs tienen una restricción que significa que después de que se envíen los registros TLS24.5 , utilizando la misma clave de sesión, la conexión se termina con el mensaje AMQ9288E. Esta restricción de GCM está activa, independientemente de la modalidad FIPS que se utilice.

    Para evitar que se produzca este error, evite utilizar cifrados TLS 1.2 GCM , habilite el restablecimiento de la clave secreta o inicie el cliente o el gestor de colas de IBM MQ con la variable de entorno `GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE` establecida. Para las bibliotecas de GSKit , debe establecer esta variable de entorno en ambos lados de la conexión y aplicarla a las conexiones de cliente a gestor de colas y al gestor de colas a las conexiones de gestor de colas. Tenga en cuenta que este valor afecta a los clientes .NET no gestionados, pero no a los clientes Java o gestionados .NET . Para obtener más información, consulte AES-GCM restricción de cifrado.

    z/OS Esta restricción no es aplicable a IBM MQ for z/OS.

Utilización de TLS 1.3 en IBM MQ

El producto da soporte a TLS 1.3 en todas las plataformas.

Los gestores de colas que se crean en IBM MQ 9.2.0 o posterior dan soporte a TLS 1.3 de forma predeterminada. Los gestores de colas migrados desde versiones anteriores de IBM MQ deben tener TLS 1.3 habilitado. Puede habilitar TLS 1.3 en gestores de colas migrados estableciendo la propiedad **AllowTLSV13**=TRUE :

- Multi Para los gestores de colas de IBM MQ for Multiplatforms , edite el archivo qm.ini y añada la propiedad **AllowTLSV13**=TRUE bajo la stanza SSL (enlace a

```
SSL:
  AllowTLSV13=TRUE
```

- z/OS Para los gestores de colas de IBM MQ for z/OS , edite el conjunto de datos QMINI especificado en el JCL de inicio del gestor de colas y añada la propiedad **AllowTLSV13**=TRUE en la stanza TransportSecurity

```
TransportSecurity:
    AllowTLSV13=TRUE
```

Cuando TLS 1.3 está habilitado, y de acuerdo con la especificación TLS 1.3, cualquier intento de comunicarse con una CipherSpecdébil, independientemente de si están habilitados en IBM MQ o no, se rechaza. Las CipherSpecs que TLS 1.3 considera que son débiles son CipherSpecs que cumplen uno o varios de los criterios siguientes:

- Utiliza el protocolo SSL 3.0.
- Utiliza RC4 o RC2 como algoritmo de cifrado.
- Tiene un tamaño de clave de cifrado (bit) igual o inferior a 112.

Estas restricciones se marcan con la nota [3] en la Tabla 1 de CipherSpecsen desuso.

Si necesita continuar utilizando estas CipherSpecs, debe inhabilitar la modalidad TLS 1.3 :

- ALW Edite el archivo qm.ini del gestor de colas y cambie el valor de la propiedad **AllowTLSV13** a:

```
SSL:
  AllowTLSV13=FALSE
```

- z/OS Edite el conjunto de datos QMINI del gestor de colas y cambie el valor de la propiedad **AllowTLSV13** por:

```
TransportSecurity:
    AllowTLSV13=FALSE
```

IBM MQ MQI client y TLS 1.3

ALW

Cuando se utiliza IBM MQ MQI client, el valor de **AllowTLSV13** se infiere a menos que se especifique explícitamente en la stanza SSL del archivo mqclient.ini que está utilizando la aplicación.

- Si las CipherSpecs débiles están habilitadas, **AllowTLSV13** se establece en FALSE y no se puede utilizar ninguna CipherSpecs de TLS 1.3.
- De lo contrario, **AllowTLSV13** se establece en TRUE y se pueden utilizar las nuevas TLS 1.3 CipherSpecs y el alias CipherSpecs .

Valores CipherSpec predeterminados habilitados en IBM MQ

En la configuración predeterminada para un nuevo gestor de colas de IBM MQ , IBM MQ proporciona soporte para los protocolos TLS 1.2 y TLS 1.3 y varios algoritmos criptográficos utilizando CipherSpecs. A efectos de compatibilidad, IBM MQ también se puede configurar para utilizar protocolos SSL 3.0 y TLS 1.0 y una serie de algoritmos de cifrado que se sabe que son débiles o susceptibles de vulnerabilidades de seguridad. La lista de CipherSpecs que están habilitadas en la configuración predeterminada puede cambiar aplicando el mantenimiento.

Es posible configurar IBM MQ para restringir o permitir el uso de CipherSpecs utilizando los controles siguientes:

- Solo permite CipherSpecs compatibles con FIPS 140-2 utilizando SSLFIPS.
- ALW Solo permitir CipherSpecs compatibles con NSA Suite B utilizando SUITEB.
- Multi Permitir una lista personalizada de CipherSpecs utilizando **AllowedCipherSpecs**.
- ALW Permitir una lista personalizada de CipherSpecs utilizando la variable de entorno **AMQ_ALLOWED_CIPHERS** .
- ALW Permitir el uso de CipherSpecs en desuso utilizando **AllowWeakCipher** o la variable de entorno **AMQ_SSL_WEAK_CIPHER_ENABLE**.
- z/OS Permitir el uso de las CipherSpecs en desuso utilizando sentencias DD en el JCL CHINIT.

Nota: Si especifica una lista personalizada de CipherSpecs utilizando **AllowedCipherSpecs** o **AMQ_ALLOWED_CIPHERS**, esto altera temporalmente la habilitación de cualquier CipherSpecs en desuso. Tenga en cuenta que al utilizar las restricciones de NSA Suite B o FIPS 140-2 en combinación con una lista de CipherSpec personalizada, debe asegurarse de que la lista personalizada solo contiene las CipherSpecs permitidas por los valores de Suite B o FIPS 140-2.

Conceptos relacionados

“Certificados digitales y compatibilidad de CipherSpec en IBM MQ” en la página 49
En este tema se proporciona información sobre cómo elegir las CipherSpecs y los certificados digitales adecuados para su política de seguridad, describiendo la relación entre las CipherSpecs y los certificados digitales en IBM MQ.

“CipherSpecs y CipherSuites” en la página 22
Los protocolos de seguridad criptográficos deben estar de acuerdo con los algoritmos utilizados por una conexión segura. CipherSpecs y CipherSuites definen combinaciones específicas de algoritmos.

“Configuración de IBM MQ para Suite B” en la página 45
IBM MQ se puede configurar para que funcione de conformidad con el estándar NSA Suite B en plataformas AIX, Linux, and Windows .

“Estándares federales de procesamiento de la información (FIPS)” en la página 35
En este tema se presenta los estándares federales de procesamiento de la información (FIPS) Cryptomodule Validation Program del US National Institute of Standards and Technology y las funciones de cifrado que se pueden utilizar en canales TLS.

Tareas relacionadas

Migración de configuraciones de seguridad existentes para utilizar un alias CipherSpe

Referencia relacionada

DEFINE CHANNEL

ALTER CHANNEL

Cambiar, Copiar y Crear canal

ALW AES-Restricción de cifrado deGCM

Una guía de las restricciones que se imponen en los cifrados AES-GCM cuando se utilizan para criptografía TLS. Estas restricciones las imponen las organizaciones IETF y NIST y requieren que no

se utilice la misma clave de sesión para transferir de forma segura más de 2 registros[24.5] TLS cuando se utilizan cifrados AES-GCM .

Para obtener más información sobre estas restricciones, consulte Sección RFC 9325 4.4 Limits on Key Usage y Sección RFC 8446 5.5.

IBM MQ no implementa la funcionalidad criptográfica directamente. En su lugar, se utilizan varias bibliotecas criptográficas diferentes para proporcionar la funcionalidad TLS y Advanced Message Security . En los sistemas operativos Windows, Linuxy AIX , la biblioteca criptográfica que IBM MQ utiliza es IBM Global Security Kit (GSKit). Para las aplicaciones, las bibliotecas C y .NET no gestionadas utilizan GSKit para la funcionalidad criptográfica. La implementación de los algoritmos de cifrado AES-GCM mediante GSKit incluye las restricciones especificadas por el grupo de estándares. Además, estas restricciones están habilitadas de forma predeterminada. Como tal, la comunicación TLS de IBM MQ , al utilizar cifrados AES-GCM , termina si se transmiten más de 2 registros TLS[24.5] utilizando la misma clave de sesión.

Nota: Esta restricción no está presente en las plataformas IBM i, IBM Z o IBM MQ for HPE NonStop o Java/JMS, las aplicaciones .NET gestionadas porque se utilizan distintas bibliotecas criptográficas y estas bibliotecas no han implementado la misma restricción.

Si un canal IBM MQ permanece en ejecución durante el tiempo suficiente para que se transmitan más de 2 registros TLS[24.5] utilizando la misma clave de sesión, la biblioteca criptográfica subyacente termina la conexión. Esto hace que el canal termine y se genere un mensaje de error AMQ9288E . Las aplicaciones que tienen su comunicación terminada de este modo reciben un código de retorno `MQRC_CONNECTION_BROKEN` de la operación IBM MQ que se estaba realizando.

La terminación de la conexión se puede realizar en cualquiera de los dos extremos de la comunicación, pero sólo en los extremos que utilizan GSKit para la funcionalidad criptográfica.

Consejos para mitigar la restricción

Algunas opciones sobre cómo evitar o manejar las comunicaciones que se terminan debido a esta restricción son las siguientes:

Utilizar clientes reconectables

Las aplicaciones se pueden configurar para que intenten automáticamente una reconexión, en caso de que falle una conexión. Esto incluye las conexiones que han terminado debido a la restricción GCM . Cuando se configura para la reconexión, la aplicación cliente se restaura automáticamente en cualquier punto de anomalía y se restaura cualquier descriptor de contexto para abrir objetos. Esto se hace sin volver al código de aplicación.

Para obtener más información, consulte Reconexión de cliente automática.

Establecer un valor de restablecimiento de clave secreta

IBM MQ se puede configurar para solicitar un restablecimiento de clave de sesión después de que se haya transferido un número configurable de bytes a través de un canal. Al alcanzar este límite, IBM MQ solicita que la capa criptográfica realice un restablecimiento de clave de sesión, lo que da como resultado una nueva clave de sesión.

Es importante tener en cuenta que el valor especificado es el número de bytes transferidos, que se relaciona con el tamaño de los mensajes enviados por IBM MQ. La restricción está en el número de registros TLS que se envían. No hay una correlación directa entre los bytes de mensajes y los registros TLS, ya que un registro TLS puede enviar un número máximo de bytes que depende de la unidad máxima de transmisión (MTU) de la red. Los mensajes que se envían que son mayores que este valor se transmiten como varios registros TLS. El valor de MTU varía entre redes. Además, existen otras razones por las que un registro TLS puede tener que enviarse fuera de la transmisión de datos de mensajes de IBM MQ , por ejemplo, IBM MQ comprobaciones de latido, alertas TLS, otros mensajes de protocolo IBM MQ . Estos registros TLS adicionales cuentan para el número máximo de registros TLS, pero no se cuentan en el valor de restablecimiento de clave secreta de IBM MQ .

El restablecimiento regular de una clave de sesión utilizando el restablecimiento de clave secreta puede impedir que el canal termine debido a la restricción AES-GCM .

Para obtener más información, consulte Restablecimiento de claves secretas SSL y TLS.

Utilizar especificaciones de cifrado TLS 1.3

Aunque la restricción AES-GCM sigue estando presente cuando se utiliza el protocolo TLS 1.3 , el protocolo TLS 1.3 da soporte a la realización automática de un restablecimiento de clave de sesión sin necesidad de interrumpir las comunicaciones TLS. Esto permite a GSKit gestionar el restablecimiento de la clave de sesión cuando sea necesario sin que IBM MQ necesite solicitar un restablecimiento de clave secreta.

Para obtener más información, consulte Utilización de TLS 1.3 en IBM MQ en "Habilitación de CipherSpecs" en la página 428.

Inhabilitar la restricción AES-GCM

Si es necesario, la restricción se puede inhabilitar estableciendo la variable de entorno **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** para inhabilitar la restricción AES-GCM . Esto permite enviar cualquier número de registros TLS utilizando la misma clave de sesión. Si elige esta mitigación, la variable de entorno debe establecerse en cada extremo de la comunicación que utiliza GSKit para las comunicaciones seguras.

Aviso: Esta opción no se recomienda ya que, después de que se hayan enviado más de 2 registros TLS[24.5] , es posible que los atacantes realicen análisis en los registros enviados para determinar la clave de sesión en uso. Una vez que se ha determinado la clave de sesión, todas las comunicaciones existentes y futuras que utilizan dicha clave de sesión se ven comprometidas.

Orden CipherSpec en el reconocimiento TLS

El orden de CipherSpecs se utiliza al elegir entre varias CipherSpecs posibles, por ejemplo, cuando se utiliza una de las CipherSpecs ANY*.

Durante un reconocimiento TLS, un cliente y un servidor intercambian las CipherSpecs y los protocolos a los que dan soporte por orden de preferencia. Una CipherSpec común que ambos lados priorizan se elige y se utiliza para la comunicación TLS. Al elegir un protocolo CipherSpec , también se tiene en cuenta la versión, por ejemplo, si un servidor lista TLS 1.2 CipherSpecs antes de TLS 1.3 CipherSpecs seguirá priorizando TLS 1.3 siempre que el cliente pueda soportarlo y tenga una TLS común 1.3 CipherSpec que se pueda utilizar.

Cuando IBM MQ está configurado para TLS, establece las CipherSpecs en el orden que se muestra en la tabla siguiente, de la más preferida a la menos preferida.

Nota: Si una CipherSpec no está habilitada a través del atributo **AllowedCipherSpecs** , no se configurará para su uso durante un reconocimiento TLS.

En el caso de que no se especifique el atributo **AllowedCipherSpecs** , se utiliza una lista predeterminada de cifrados habilitados, indicada por la tabla siguiente.

Tabla 78. CipherSpecs de IBM MQ 9.2.0

Plataforma	CipherSpec	Protocolo	Código hexadecimal	Habilitado de forma predeterminada
Todo	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sí
Todo	TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sí
Todo	TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sí
	TLS_AES_128_CCM_SHA256	TLS 1.3	1304	Sí

Tabla 78. CipherSpecs de IBM MQ 9.2.0 (continuación)				
Plataforma	**CipherSpec**	**Protocolo**	**Código hexadecimal**	**Habilitado de forma predeterminada**
	TLS_AES_128_CCM_8_SHA256	TLS 1.3	1305	Sí
Todo	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sí
	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	C02C	Sí
Todo	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sí
Todo	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sí
Todo	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sí
Todo	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sí
Todo	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sí
	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	C02B	Sí
Todo	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sí
Todo	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sí
Todo	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sí
Todo	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sí
	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	C008	No
	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	C012	No
	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	0005	No

Tabla 78. CipherSpecs de IBM MQ 9.2.0 (continuación)				
Plataforma	**CipherSpec**	**Protocolo**	**Código hexadecimal**	**Habilitado de forma predeterminada**
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	C007	No
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	C011	No
Todo	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
Todo	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	No
Todo	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	No
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	0004	No
Todo	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

Tabla 78. CipherSpecs de IBM MQ 9.2.0 (continuación)				
Plataforma	**CipherSpec**	**Protocolo**	**Código hexadecimal**	**Habilitado de forma predeterminada**
Todo	TRIPLE_DES_SHA_US	SSL v3	000A	No
Todo	RC4_SHA_US	SSL v3	0005	No
Todo	RC4_MD5_US	SSL v3	0004	No
Todo	DES_SHA_EXPORT	SSL v3	0009	No
Todo	RC4_MD5_EXPORT	SSL v3	0003	No
Todo	RC2_MD5_EXPORT	SSL v3	0006	No
Todo	NULL_SHA	SSL v3	0002	No
Todo	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 se ha creado ordenando los protocolos con la lista predeterminada proporcionada por la biblioteca criptográfica utilizada por IBM MQ en z/OS y es coherente entre z/OS y las plataformas distribuidas.

cambiar el orden

Si se desea un orden diferente, se puede proporcionar un nuevo orden de CipherSpecs utilizando el atributo **AllowedCipherSpecs** de la stanza SSL en IBM MQ for Multiplatforms z/OS, o la stanza TransportSecurity en IBM MQ for z/OS, con las reglas siguientes:

- Siempre se utilizan versiones de protocolo más altas, independientemente de su posición en la lista.
- Las CipherSpecs inhabilitadas se vuelven a habilitar si se proporcionan en la lista.
- El orden de lista del servidor TLS tiene una prioridad más alta que el cliente TLS.
- Cuando TLS 1.3 está habilitado, determinadas CipherSpecs no están soportadas.

Por ejemplo, en IBM MQ for Multiplatforms, si se configura lo siguiente en el gestor de colas:

```
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 y en IBM MQ for z/OS, si se configura lo siguiente en el gestor de colas:

```
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
```

y a continuación:

- Un cliente que se conecte con ANY_TLS12 probablemente utilizará TLS 1.2 CipherSpec TLS_RSA_WITH_AES_128_GCM_SHA256.
- Un cliente que se conecte con ANY_TLS12_OR_HIGHER probablemente utilizará la TLS 1.3 CipherSpec TLS_AES_128_GCM_SHA256 (suponiendo que el cliente soporte TLS 1.3).
- Un cliente que se conecte con TLS 1.0 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA utilizará esa CipherSpec.

Versiones anteriores de IBM MQ

Antes de IBM MQ 9.2.0, se utilizaba el siguiente orden de CipherSpecs :

Tabla 79. CipherSpecs antes de IBM MQ 9.2.0

Plataforma	CipherSpec	Protocolo	Habilitado de forma predeterminada
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
Todo	RC4_SHA_US	SSL v3	No
Todo	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	No
Todo	RC4_MD5_US	SSL v3	No
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	No
Todo	TRIPLE_DES_SHA_US	SSL v3	No
Todo	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	No
ALW	DES_SHA_EXPORT1024	SSL v3	No
Todo	RC4_56_SHA_EXPORT1024	SSL v3	No
Todo	RC4_MD5_EXPORT	SSL v3	No
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	No
Todo	RC2_MD5_EXPORT	SSL v3	No
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	No
Todo	DES_SHA_EXPORT	SSL v3	No
Todo	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	No

Tabla 79. CipherSpecs antes de IBM MQ 9.2.0 (continuación)			
Plataforma	**CipherSpec**	**Protocolo**	**Habilitado de forma predeterminada**
Todo	NULL_SHA	SSL v3	No
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	No
Todo	NULL_MD5	SSL v3	No
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
Todo	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	Sí
Todo	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	Sí
Todo	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	No
Todo	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	Sí
Todo	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	Sí
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
Todo	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	Sí
Todo	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	Sí
Todo	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	Sí
Todo	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	Sí
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	Sí
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	Sí

Tabla 79. CipherSpecs antes de IBM MQ 9.2.0 (continuación)			
Plataforma	**CipherSpec**	**Protocolo**	**Habilitado de forma predeterminada**
Todo	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	Sí
Todo	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	Sí
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	No
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	Sí
Multi	TLS_AES_256_GCM_SHA384	TLS 1.3	Sí
Multi	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	Sí
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	Sí
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	Sí

Importante: A partir del 23rd de julio de 2020, el siguiente atributo AllowedCipherSpecs sólo habilita las CipherSpecs que están habilitadas actualmente de forma predeterminada. Sin embargo, debe verificar las CipherSpecs habilitadas por el siguiente atributo AllowedCipherSpecs con datos actuales, para asegurarse de que las CipherSpecs que han quedado en desuso desde esta fecha no se vuelvan a habilitar inadvertidamente.

Si necesita volver a este orden de CipherSpecs, puede hacerlo utilizando el siguiente valor de atributo de stanza **AllowedCipherSpecs** SSL/TransportSecurity :

```
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
```

Proporcionar una lista personalizada de CipherSpecs ordenadas y habilitadas en IBM MQ for Multiplatforms

Multi

Puede proporcionar un conjunto alternativo de CipherSpecs que están habilitadas, y en su orden de preferencia, para su uso con canales IBM MQ , utilizando la variable de entorno ALW **AMQ_ALLOWED_CIPHERS** o el atributo de stanza SSL **AllowedCipherSpecs** del archivo `.ini` . Es posible que desee utilizar este valor por una de las razones siguientes:

- Para impedir que los escuchas de IBM MQ acepten solicitudes de inicio de canal de entrada, a menos que utilicen una de las CipherSpecsdenominadas.
- Para cambiar el orden de prioridad de las CipherSpecs que se utilizan en un reconocimiento TLS.

Esta funcionalidad se puede utilizar para controlar las CipherSpecs que se incluyen en las ANY* CipherSpecs.

La variable de entorno **AMQ_ALLOWED_CIPHERS** o el atributo **AllowedCipherSpecs** de stanza SSL acepta:

- Un nombre de CipherSpec único.
- Una lista separada por comas de nombres de CipherSpec que se van a volver a habilitar.
- El valor especial de ALL, que representa todas las CipherSpecs.

Nota: No debe habilitar **ALL** CipherSpecs, ya que esto habilitará los protocolos SSL 3.0 y TLS 1.0 y un gran número de algoritmos criptográficos débiles.

Si se ha configurado este valor, sustituye la lista de CipherSpec predeterminada y hace que IBM MQ ignore los valores de cifrado débiles en desuso (consulte más abajo):

- Los escuchas de IBM MQ sólo aceptan propuestas SSL/TLS que utilizan una de las CipherSpecscon nombre.
- Los canales IBM MQ sólo permiten un valor SSLCIPH en blanco, o una de las CipherSpecscon nombre.
- La finalización del separador **runmqsc** de los valores SSLCIPH restringe los valores de terminación a una de las CipherSpecs especificadas.

Por ejemplo, si solo desea permitir que los canales se definan/alteren y que los escuchas acepten ECDHE_RSA_AES_128_GCM_SHA256 o ECDHE_ECDSA_AES_256_GCM_SHA384, podría establecer lo siguiente en el archivo qm.ini:

```
SSL:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

Además, las CipherSpecs de esta lista se utilizarán para determinar la prioridad de las CipherSpecs utilizadas durante un reconocimiento TLS. Por ejemplo, si especifica una lista de TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256 , es probable que, durante el reconocimiento, se elija TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec a través de TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec si un cliente se conecta especificando ambas CipherSpecs, es decir, un cliente que se conecta con ANY_TLS12.

Tenga en cuenta que los cifrados utilizados por canales AMQP o MQTT se pueden restringir utilizando los valores del archivo java.security.

Proporcionar una lista personalizada de CipherSpecs ordenadas y habilitadas en IBM MQ for z/OS

Es posible proporcionar un conjunto alternativo de CipherSpecs que están habilitadas, y en su orden de preferencia, para su uso con canales IBM MQ , utilizando el atributo de stanza **AllowedCipherSpecs** TransportSecurity de El conjunto de datos QMINI. Es posible que desee hacerlo por una de las razones siguientes:

- Para impedir que los escuchas de IBM MQ acepten solicitudes de inicio de canal de entrada, a menos que utilicen una de las CipherSpecsdenominadas.
- Para cambiar el orden de prioridad de las CipherSpecs que se utilizan en un reconocimiento TLS.

Puede utilizar esta funcionalidad para controlar las CipherSpecs que se incluyen en las ANY* CipherSpecs. El atributo **AllowedCipherSpecs** acepta:

- Un nombre de CipherSpec único.

- Una lista separada por comas de nombres de CipherSpec que se van a volver a habilitar.
- El valor especial de ALL, que representa todas las CipherSpecs.

  Nota: No debe habilitar **ALL** CipherSpecs, ya que esto habilitará los protocolos SSL 3.0 y TLS 1.0 y un gran número de algoritmos criptográficos débiles. Si configura este valor, altera temporalmente la lista CipherSpec predeterminada y hace que IBM MQ ignore los valores de desuso de cifrado débil; consulte “Habilitación de CipherSpecs en desuso en z/OS” en la página 448.

Los escuchas de IBM MQ sólo aceptan propuestas SSL/TLS que utilizan una de las CipherSpecs y los canales IBM MQ con nombre solo permiten un valor SSLCIPH en blanco o una de las CipherSpecscon nombre.

Por ejemplo, si sólo desea permitir que los canales se definan/alteren y los escuchas acepten ECDHE_RSA_AES_128_GCM_SHA256 o ECDHE_RSA_AES_256_GCM_SHA384 , puede establecer lo siguiente:

```
TransportSecurity:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                     ECDHE_RSA_AES_256_GCM_SHA384
```

Además, las CipherSpecs de esta lista se utilizan para determinar la prioridad de las CipherSpecs utilizadas durante un reconocimiento TLS. Por ejemplo, si especifica una lista de TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256 es probable que, durante el reconocimiento, TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec se elija entre TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec si un cliente se conecta a ambos Cipher_$tag13.

Deprecated CipherSpecs en desuso

Una lista de CipherSpecs en desuso que puede utilizar con IBM MQ si es necesario.

En la siguiente tabla se listan las CipherSpecs en desuso que puede utilizar con el soporte TLS de IBM MQ.

Tabla 80. CipherSpecs en desuso que puede volver a habilitar para su uso con IBM MQ

Soporte de plataforma “1” en la página 447	Nombre de CipherSpec	Código hexadecimal	Protocolo utilizado	Integridad de datos	Algoritmo de cifrado (bits de cifrado)	FIPS “2” en la página 447	Suite B	Actualizar cuando esté en desuso
CipherSpecs para SSL 3.0								
IBM i	AES_SHA_US “3” en la página 447	002F	SSL 3.0	SHA-1	AES (128)	No	No	9.0.0.0
Todo	DES_SHA_EXPORT “3” en la página 447 “4” en la página 447 “5” en la página 447	0009	SSL 3.0	SHA-1	DES (56)	No	No	9.0.0.0
ALW	DES_SHA_EXPORT1024 “3” en la página 447 “6” en la página 447	0062	SSL 3.0	SHA-1	DES (56)	No	No	9.0.0.0
ALW	FIPS_WITH_DES_CBC_SHA “3” en la página 447	FEFE	SSL 3.0	SHA-1	DES (56)	No “7” en la página 447	No	9.0.0.0

Tabla 80. CipherSpecs en desuso que puede volver a habilitar para su uso con IBM MQ (continuación)

Soporte de plataforma "1" en la página 447	Nombre de CipherSpec	Código hexadecimal	Protocolo utilizado	Integridad de datos	Algoritmo de cifrado (bits de cifrado)	FIPS "2" en la página 447	Suite B	Actualizar cuando esté en desuso
ALW	FIPS_WITH_3DES_EDE_CBC_SHA "3" en la página 447	FEFF	SSL 3.0	SHA-1	3DES (168)	No"8" en la página 447	No	9.0.0.1 y 9.0.1
Todo	NULL_MD5 "3" en la página 447	0001	SSL 3.0	MD5	Ninguna	No	No	9.0.0.1
Todo	NULL_SHA "3" en la página 447	0002	SSL 3.0	SHA-1	Ninguna	No	No	9.0.0.1
Todo	RC2_MD5_EXPORT "3" en la página 447 "4" en la página 447 "5" en la página 447	0006	SSL 3.0	MD5	RC2 (40)	No	No	9.0.0.0
Todo	RC4_MD5_EXPORT "4" en la página 447 "3" en la página 447	0003	SSL 3.0	MD5	RC4 (40)	No	No	9.0.0.0
Todo	RC4_MD5_US "3" en la página 447	0004	SSL 3.0	MD5	RC4 (128)	No	No	9.0.0.0
Todo	RC4_SHA_US "3" en la página 447 "5" en la página 447	0005	SSL 3.0	SHA-1	RC4 (128)	No	No	9.0.0.0
ALW	RC4_56_SHA_EXPORT1024 "3" en la página 447 "6" en la página 447	0064	SSL 3.0	SHA-1	RC4 (56)	No	No	9.0.0.0
Todo	TRIPLE_DES_SHA_US "3" en la página 447 "5" en la página 447	000A	SSL 3.0	SHA-1	3DES (168)	No	No	9.0.0.1 y 9.0.1
CipherSpecs para TLS 1.0								
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5 "3" en la página 447	0006	TLS 1.0	MD5	RC2 (40)	No	No	9.0.0.0
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5"3" en la página 447 "4" en la página 447	0003	TLS 1.0	MD5	RC4 (40)	No	No	9.0.0.0
Todo	TLS_RSA_WITH_DES_CBC_SHA "3" en la página 447	0009	TLS 1.0	SHA-1	DES (56)	No"9" en la página 447	No	9.0.0.0
IBM i	TLS_RSA_WITH_NULL_MD5 "3" en la página 447	0001	TLS 1.0	MD5	Ninguna	No	No	9.0.0.1
IBM i	TLS_RSA_WITH_NULL_SHA "3" en la página 447	0002	TLS 1.0	SHA-1	Ninguna	No	No	9.0.0.1
IBM i	TLS_RSA_WITH_RC4_128_MD5 "3" en la página 447	0004	TLS 1.0	MD5	RC4 (128)	No	No	9.0.0.0
z/OS ALW	TLS_RSA_WITH_AES_128_CBC_SHA "10" en la página 447	002F	TLS 1.0	SHA-1	AES (128)	Sí	No	9.0.5
z/OS ALW	TLS_RSA_WITH_AES_256_CBC_SHA "6" en la página 447 "10" en la página 447	0035	TLS 1.0	SHA-1	AES (256)	Sí	No	9.0.5

Tabla 80. CipherSpecs en desuso que puede volver a habilitar para su uso con IBM MQ (continuación)

Soporte de plataforma "1" en la página 447	Nombre de CipherSpec	Código hexadecimal	Protocolo utilizado	Integridad de datos	Algoritmo de cifrado (bits de cifrado)	FIPS "2" en la página 447	Suite B	Actualizar cuando esté en desuso
Todo	TLS_RSA_WITH_3DES_EDE_CBC_SHA	000A	TLS 1.0	SHA-1	3DES (168)	Sí	No	9.0.0.1 y 9.0.1
CipherSpecs para TLS 1.2								
ALW	ECDHE_ECDSA_NULL_SHA256 "3" en la página 447	C006	TLS 1.2	SHA-1	Ninguna	No	No	9.0.0.1
ALW	ECDHE_ECDSA_RC4_128_SHA256 "3" en la página 447	C007	TLS 1.2	SHA-1	RC4 (128)	No	No	9.0.0.0
ALW IBM i	ECDHE_RSA_NULL_SHA256 "3" en la página 447	C010	TLS 1.2	SHA-1	Ninguna	No	No	9.0.0.1
ALW IBM i	ECDHE_RSA_RC4_128_SHA256 "3" en la página 447	C011	TLS 1.2	SHA-1	RC4 (128)	No	No	9.0.0.0
ALW	TLS_RSA_WITH_NULL_NULL "3" en la página 447	0000	TLS 1.2	Ninguna	Ninguna	No	No	9.0.0.1
Todo	TLS_RSA_WITH_NULL_SHA256 "3" en la página 447	003B	TLS 1.2	SHA-256	Ninguna	No	No	9.0.0.1
ALW	TLS_RSA_WITH_RC4_128_SHA256 "3" en la página 447	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)	Sí	No	9.0.0.1 y 9.0.1
ALW IBM i	ECDHE_RSA_3DES_EDE_CBC_SHA256	C012	TLS 1.2	SHA-1	3DES (168)	Sí	No	9.0.0.1 y 9.0.1

Tabla 80. CipherSpecs en desuso que puede volver a habilitar para su uso con IBM MQ (continuación)

Soporte de plataforma "1" en la página 447	Nombre de CipherSpec	Código hexadecimal	Protocolo utilizado	Integridad de datos	Algoritmo de cifrado (bits de cifrado)	FIPS "2" en la página 447	Suite B	Actualizar cuando esté en desuso

Notas:

1. Para obtener una lista de las plataformas cubiertas por cada icono de plataforma, consulte Iconos utilizados en la documentación del producto.
2. Especifica si la CipherSpec tiene el certificado FIPS en una plataforma certificada con FIPS. Consulte Federal Information Processing Standards (FIPS - Estándares federales de procesamiento de información) para obtener una explicación de FIPS.
3. ALW Estas CipherSpecs están inhabilitadas cuando TLS 1.3 está habilitado (a través de la propiedad AllowTLSV13 en `qm.ini`).

   z/OS Los gestores de colas creados en IBM MQ for z/OS 9.2.0 o posterior permiten TLS 1.3 de forma predeterminada, lo que inhabilita estas CipherSpecs. Puede habilitar estas CipherSpecs, si es necesario, desactivando TLS V1.3. Esto se lleva a cabo añadiendo **AllowTLSV13**=*FALSE* a la stanza TransportSecurity del conjunto de datos QMINI en el JCL del gestor de colas. Los gestores de colas migrados a IBM MQ for z/OS 9.2.0 de una versión anterior no tienen TLS 1.3 habilitado de forma predeterminada y por lo tanto tienen habilitadas estas CipherSpecs.
4. El tamaño máximo de la clave de reconocimiento es de 512 bits. Si cualquiera de los certificados intercambiados durante el reconocimiento SSL tiene un tamaño de clave mayor de 512 bits, se genera una clave temporal de 512 bits para poder utilizarla durante el reconocimiento.
5. IBM MQ classes for Java o IBM MQ classes for JMS ya no soportan estas CipherSpecs. Para obtener más información, consulte CipherSpecs y CipherSuites SSL/TLS en IBM MQ classes for Java o CipherSpecs y CipherSuites SSL/TLS en IBM MQ classes for JMS.
6. El tamaño de clave de reconocimiento es de 1024 bits.
7. Deprecated Esta CipherSpec obtuvo el certificado FIPS 140-2 antes del 19 mayo de 2007. El nombre `FIPS_WITH_DES_CBC_SHA` es histórico y refleja el hecho de que este CipherSpec era anteriormente (pero ya no lo es) compatible con FIPS. Esta CipherSpec está en desuso y su uso no se recomienda.
8. Deprecated El nombre `FIPS_WITH_3DES_EDE_CBC_SHA` es histórico y refleja el hecho de que este CipherSpec era anteriormente (pero ya no lo es) compatible con FIPS. Esta CipherSpec está en desuso.
9. Esta CipherSpec obtuvo el certificado FIPS 140-2 antes del 19 mayo de 2007.
10. Volver a habilitar sólo estas CipherSpecs no requiere el uso de la sentencia CSQXWEAK DD.

Habilitación de CipherSpecs en desuso en IBM MQ for Multiplatforms

Multi

De forma predeterminada, no está permitido especificar una CipherSpec en desuso en una definición de canal. Si intenta especificar una CipherSpec en desuso en IBM MQ for Multiplatforms, recibirá el mensaje AMQ8242: definición SSLCIPH errónea y PCF devuelve `MQRCCF_SSL_CIPHER_SPEC_ERROR`.

No puede iniciar un canal con una CipherSpec en desuso. Si intenta hacerlo con una CipherSpecen desuso, el sistema devuelve `MQCC_FAILED (2)`, junto con un **Reason** de `MQRC_SSL_INITIALIZATION_ERROR (2393)` al cliente.

Puede volver a habilitar una o más de las CipherSpecs en desuso para definir canales, en tiempo de ejecución en el servidor, estableciendo la variable de entorno **AMQ_SSL_WEAK_CIPHER_ENABLE**.

La variable de entorno **AMQ_SSL_WEAK_CIPHER_ENABLE** acepta:

- Un solo nombre de CipherSpec o
- Una lista separada por comas de nombres CipherSpec para volver a habilitar, o
- El valor especial de ALL, que representa todas las CipherSpecs.

Atención: Aunque ALL es una opción válida, debe utilizarla **solo** en una situación específica que la empresa requiera, ya que al volver a habilitar ALL CipherSpecs habilita los protocolos SSL 3.0 y TLS 1.0 , así como un gran número de algoritmos criptográficos débiles.

Por ejemplo, si desea volver a habilitar ECDHE_RSA_RC4_128_SHA256, establezca la siguiente variable de entorno:

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

o, como alternativa, cambie la stanza SSL en el archivo qm.ini estableciendo:

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

Habilitación de CipherSpecs en desuso en z/OS

De forma predeterminada, no está permitido especificar una CipherSpec en desuso en una definición de canal. Si intenta especificar una CipherSpec en desuso en z/OS, recibirá el mensaje CSQM102E, el mensaje CSQX616Eo CSQX674E.

Siga las instrucciones listadas en esta sección si recibe alguno de estos mensajes y su empresa necesita volver a habilitar el uso de CipherSpecsdébiles.

Atención: En las instrucciones siguientes, para que las sentencias de definición ficticia (DD) entren en vigor, SSLTASKS debe ser un valor distinto de cero. Si esto requiere un cambio en SSLTASKS, debe reciclar el iniciador de canal.

En IBM MQ for z/OS, el método actual para controlar las CipherSpecs débiles o rotas es el siguiente:

- Si desea volver a habilitar el uso de CipherSpecsdébiles, puede hacerlo añadiendo una sentencia de definición de datos ficticia (DD) denominada CSQXWEAK al JCL del iniciador de canal. Si se especifica por sí solo, esto sólo habilita CipherSpecs débiles asociadas con el protocolo TLS 1.2 ; por ejemplo:

  ```
  //CSQXWEAK DD DUMMY
  ```

  Nota: No todas las CipherSpecs en desuso requieren el uso de esta sentencia DD, consulte la nota 10 de la tabla anterior.
- Si desea volver a habilitar el uso de SSLv3 CipherSpecs, puede hacerlo añadiendo también una sentencia DD ficticia denominada CSQXSSL3 al JCL del iniciador de canal. Todas las SSLv3 CipherSpecs se consideran **débiles**, por lo que también debe especificar CSQXWEAK:

  ```
  //CSQXSSL3 DD DUMMY
  ```
- Si desea volver a habilitar las CipherSpecsTLS V1 en desuso, puede hacerlo añadiendo una sentencia DD ficticia denominada TLS10ON (active TLS V1.0) al JCL del iniciador de canal. Si se especifica por sí solo, esto habilita CipherSpecs fuertes asociadas con el protocolo TLS 1.0 :

  ```
  //TLS10ON DD DUMMY
  ```

  Si se especifica con CSQXWEAK , esto también habilita **Weak** CipherSpecs asociadas con TLS 1.0.

- Si desea desactivar explícitamente TLS V1 CipherSpecsen desuso, puede hacerlo añadiendo una sentencia DD ficticia denominada TLS10OFF (desactivar TLS V1.0) al JCL del iniciador de canal; por ejemplo:

```
//TLS10OFF DD DUMMY
```

Si sólo desea negociar con el escucha utilizando las especificaciones de cifrado listadas en la lista de especificaciones de cifrado predeterminadas de **System SSL** , debe definir la siguiente sentencia DD en el JCL de CHINIT:

```
JCL: //GSKDCIPS DD DUMMY
```

Importante: Para IBM MQ for z/OS 9.2.0 y posteriores, las tarjetas DD listadas anteriormente y el valor de **AllowTLSV13** se tienen en cuenta al visualizar mensajes durante el inicio del iniciador de canal para indicar qué protocolos están habilitados y cuáles no. Por lo tanto, incluso si se especifica una de las tarjetas DD listadas anteriormente, podría significar que, debido a una combinación de estos valores, no se puede habilitar un determinado protocolo con otro protocolo. Por ejemplo, el protocolo SSL 3.0 no está permitido si TLS 1.3 está habilitado.

Existen mecanismos alternativos que se pueden utilizar para volver a habilitar de forma forzada las CipherSpecsdébiles y el soporte de SSLv3 , si el cambio de definición de datos no es adecuado. Póngase en contacto con el servicio de IBM para obtener información adicional.

Conceptos relacionados

“Certificados digitales y compatibilidad de CipherSpec en IBM MQ” en la página 49
En este tema se proporciona información sobre cómo elegir las CipherSpecs y los certificados digitales adecuados para su política de seguridad, describiendo la relación entre las CipherSpecs y los certificados digitales en IBM MQ.

Referencia relacionada

DEFINE CHANNEL

ALTER CHANNEL

Relación entre los valores de alias CipherSpec

Esta información describe el comportamiento esperado con diferentes combinaciones de alias CipherSpecs en configuraciones de cliente y servidor. Aquí, un cliente hace referencia a la entidad que inicia la comunicación, por ejemplo, una aplicación cliente o un canal emisor de gestor de colas, y el servidor hace referencia a la entidad que recibe la comunicación del cliente, por ejemplo, un canal de conexión con el servidor o un canal receptor.

Protocolo mínimo frente a protocolo fijo CipherSpecs

IBM MQ da soporte a dos tipos distintos de CipherSpecs:

Protocolo mínimo
: Las CipherSpecs de protocolo mínimo son aquellas que no establecen un límite superior, por ejemplo ANY, ANY_TLS12_OR_HIGHER o ANY_TLS13_OR_HIGHER.

Protocolo fijo
: Las CipherSpecs de protocolo fijo son aquellas que identifican un protocolo específico, por ejemplo ANY_TLS12 y ANY_TLS13, o un algoritmo específico como ECDHE_ECDSA_3DES_EDE_CBC_SHA256.

Las CipherSpecs de protocolo mínimo y fijo están soportadas en todas las plataformas.

Para maximizar la simplicidad de la configuración mientras se mantiene la seguridad, se recomienda el uso del **protocolo mínimo** CipherSpecs en ambos lados del canal. Esto permite que las comunicaciones puedan dar soporte automáticamente a una versión de protocolo TLS superior, y utilizarlo, cuando ambos lados dan soporte a una nueva versión sin necesidad de cambiar la configuración de ambos lados.

Si se utiliza un **protocolo mínimo** CipherSpec en el lado de inicio, pero un **protocolo fijo** CipherSpec en el lado de recepción podría dar como resultado que se rechazara la conexión, y

- Multi Se están emitiendo los mensajes AMQ9631 y AMQ9641 .
- z/OS Mensajes CSQX631E y CSQX641E que se están emitiendo.

Las tablas siguientes muestran la relación entre los distintos valores de alias CipherSpec y el resultado esperado. Tabla 81 en la página 450 muestra el comportamiento esperado cuando TLS 1.3 no está habilitado en el cliente, el servidor o ambos. La Tabla 82 en la página 450 muestra el comportamiento esperado cuando TLS 1.3 está habilitado tanto en el cliente como en el servidor. En ambos casos, las CipherSpecs para el cliente se muestran en el eje Y de la tabla, y las CipherSpecs para el servidor se muestran en el eje X de la tabla.

Nota: En las tablas siguientes, las celdas marcadas como *Probable que falle* indican la posibilidad de conflicto cuando se especifica un **protocolo mínimo** CipherSpec para una parte de una conexión y un CipherSpec específico (**protocolo fijo**) para otra parte.

Por ejemplo, supongamos que el cliente y el servidor están establecidos para utilizar ANY CipherSpec, y el canal de servidor está establecido para utilizar una CipherSpecespecífica:

- Si la CipherSpec soportada más fuerte para el cliente y el servidor coincide con la CipherSpec específica configurada en el canal, el reconocimiento TLS se resuelve correctamente.
- Sin embargo, si hay una CipherSpec más fuerte que el soporte del cliente y del servidor, el reconocimiento TLS se resuelve utilizando esto, aunque no coincida con la CipherSpec especificada en el canal, y el reconocimiento TLS falla.

Tabla 81. Comportamiento esperado cuando TLS 1.3 no está habilitado en el cliente, el servidor o ambos

	Servidor			
Cliente	**TLS específico 1.2 CipherSpec**	**CUALQUIERA**	**ANY_ TLS12**	**ANY_TLS12_ OR_SUPERIOR**
TLS específico 1.2 CipherSpec	Conecta	Conecta	Conecta	Conecta
cualquiera	*Es probable que falle*	Conecta	Conecta	Conecta
ANY_ TLS12	*Es probable que falle*	Conecta	Conecta	Conecta
ANY_TLS12_ OR_SUPERIOR	*Es probable que falle*	Conecta	Conecta	Conecta

Tabla 82. Comportamiento esperado cuando TLS 1.3 está habilitado tanto en el cliente como en el servidor

	Servidor						
Cliente	**TLS específico 1.2 CipherSpec**	**TLS específico 1.3 CipherSpec**	**CUALQUIERA**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ OR_SUPERIOR**	**ANY_TLS13_ OR_SUPERIOR**
TLS específico 1.2 CipherSpec	Conecta	**Fallos**	Conecta	Conecta	**Fallos**	Conecta	**Fallos**
TLS específico 1.3 CipherSpec	**Fallos**	Conecta	Conecta	**Fallos**	Conecta	Conecta	Conecta

Tabla 82. Comportamiento esperado cuando TLS 1.3 está habilitado tanto en el cliente como en el servidor (continuación)

	Servidor						
Cliente	**TLS específico 1.2 CipherSpec**	**TLS específico 1.3 CipherSpec**	**CUALQUIERA**	**ANY_TLS12**	**ANY_TLS13**	**ANY_TLS12_OR_SUPERIOR**	**ANY_TLS13_OR_SUPERIOR**
cualquiera	**Fallos**	*Es probable que falle*	Conecta	**Fallos**	Conecta	Conecta	Conecta
ANY_TLS12	*Es probable que falle*	**Fallos**	Conecta	Conecta	**Fallos**	Conecta	**Fallos**
ANY_TLS13	**Fallos**	*Es probable que falle*	Conecta	**Fallos**	Conecta	Conecta	Conecta
ANY_TLS12_OR_SUPERIOR	**Fallos**	*Es probable que falle*	Conecta	**Fallos**	Conecta	Conecta	Conecta
ANY_TLS13_OR_SUPERIOR	**Fallos**	*Es probable que falle*	Conecta	**Fallos**	Conecta	Conecta	Conecta

Conceptos relacionados

"Certificados digitales y compatibilidad de CipherSpec en IBM MQ" en la página 49
En este tema se proporciona información sobre cómo elegir las CipherSpecs y los certificados digitales adecuados para su política de seguridad, describiendo la relación entre las CipherSpecs y los certificados digitales en IBM MQ.

"CipherSpecs y CipherSuites" en la página 22
Los protocolos de seguridad criptográficos deben estar de acuerdo con los algoritmos utilizados por una conexión segura. CipherSpecs y CipherSuites definen combinaciones específicas de algoritmos.

"Habilitación de CipherSpecs" en la página 428
Habilite una CipherSpec utilizando el parámetro **SSLCIPH** en el mandato **DEFINE CHANNEL** o **ALTER CHANNEL** MQSC.

Tareas relacionadas

Migración de configuraciones de seguridad existentes para utilizar la CipherSpec ANY_TLS12_OR_HIGHER

Obtención de información sobre CipherSpecs utilizando IBM MQ Explorer

Puede utilizar IBM MQ Explorer para visualizar descripciones de CipherSpecs.

Utilice el procedimiento siguiente para obtener información acerca de las CipherSpecs que aparecen en la "Habilitación de CipherSpecs" en la página 428:

1. Abra IBM MQ Explorer y expanda la carpeta **Gestores de colas**.
2. Asegúrese de que ha iniciado el gestor de colas.
3. Seleccione el gestor de colas con el que desea trabajar y pulse **Canales**.
4. Pulse con el botón derecho del ratón el canal con el que desee trabajar y seleccione **Propiedades**.
5. Seleccione la página de propiedades **SSL**.
6. Seleccione en la lista la CipherSpec con la que desea trabajar. Se visualiza una descripción en la ventana que hay debajo de la lista.

z/OS IBM i Alternativas para especificar las CipherSpecs

En aquellas plataformas en las que el sistema operativo da soporte a TLS, es posible que el sistema dé soporte a nuevas CipherSpecs que no figuran en la “Habilitación de CipherSpecs” en la página 428.

Puede especificar una nueva CipherSpec con el parámetro SSLCIPH, pero el valor que suministre dependerá de la plataforma. En todos los casos, la especificación debe corresponder a una TLS CipherSpec que es válida y, también, compatible con la versión de TLS que está ejecutando el sistema.

Nota: Esta sección no se aplica a los sistemas AIX, Linux, and Windows , porque las CipherSpecs se proporcionan con el producto IBM MQ , por lo que las nuevas CipherSpecs no pasan a estar disponibles después del envío.

IBM i IBM i

Una serie de dos caracteres que representa un valor hexadecimal.

Si desea más información sobre los valores permitidos, consulte el punto tres en las sección Notas de uso de Establecer información de carácter para una sesión segura.

Atención: No debe especificar valores de cifrado hexadecimal en **SSLCIPH**, porque no está claro a partir del valor qué cifrado se utilizará y la opción de qué protocolo se utilizará es indeterminada. El uso de los valores de cifrado hexadecimal puede llevar a errores de discrepancia de CipherSpec.

Puede utilizar el mandato **CHGMQMCHL** o el mandato **CRTMQMCHL** para especificar el valor, por ejemplo:

```
CRTMQMCHL CHLNAME(' channel name ') SSLCIPH(' hexadecimal value ')
```

También puede utilizar el mandato MQSC de **ALTER QMGR** para establecer el parámetro **SSLCIPH** .

z/OS z/OS

Una serie de cuatro caracteres que representa un valor hexadecimal. Los códigos hexadecimales se corresponden con los valores definidos en el protocolo TLS.

Para obtener más información, consulte Definiciones de suite de cifrado donde hay una lista de todas las especificaciones de cifrado TLS 1.0, TLS 1.2y TLS 1.3 soportadas en forma de códigos hexadecimales de 4 dígitos.

Nota: Deprecated Para utilizar una CipherSpecdébil o una CipherSpec perteneciente a un protocolo en desuso, como SSL V3.0 o TLS 1.0, debe especificar la tarjeta DD relevante en el JCL de inicio del iniciador de canal. Consulte “CipherSpecs en desuso” en la página 444 para obtener más información.

Consideraciones sobre los clústeres de IBM MQ

Con los clústeres de IBM MQ, es más seguro utilizar los nombres de CipherSpec de la “Habilitación de CipherSpecs” en la página 428. Si utiliza una especificación alternativa, tenga en cuenta que la especificación puede no ser válida en otras plataformas. Para obtener más información, consulte “SSL/TLS y clústeres” en la página 491.

Especificación de una CipherSpec para un IBM MQ MQI client

Dispone de tres opciones para especificar una CipherSpec para un IBM MQ MQI client.

Estas opciones son las siguientes:

- Utilizar una tabla de definiciones de canal
- Utilizando el campo SSLCipherSpec en la estructura MQCD, en MQCD_VERSION_7 o superior, en una llamada MQCONNX.
- Utilizar Active Directory (en sistemas Windows con soporte para Active Directory)

Especificación de una CipherSuite con IBM MQ classes for Java y IBM MQ classes for JMS

IBM MQ classes for Java y IBM MQ classes for JMS especifican las CipherSuites de forma diferente de otras plataformas.

Para obtener información sobre cómo especificar una CipherSuite con IBM MQ classes for Java, consulte Soporte de TLS (seguridad de la capa de transporte) para Java

Para obtener información sobre cómo especificar una CipherSuite con IBM MQ classes for JMS, consulte Utilización de TLS (seguridad de la capa de transporte) con IBM MQ classes for JMS

Especificación de una CipherSpec para un IBM MQ.NET

En IBM MQ.NET, puede especificar la CipherSpec utilizando la clase MQEnvironment o utilizando MQC.SSL_CIPHER_SPEC_PROPERTY en la tabla hash de las propiedades de conexión.

Para obtener información acerca de cómo especificar una CipherSpec para el cliente .NET no gestionado, consulte Habilitación de TLS para el cliente no gestionado.NET

Para obtener información acerca de cómo especificar una CipherSpec para el cliente .NET gestionado, consulte Soporte de CipherSpec para el cliente .NETgestionado

z/OS Uso de AT-TLS con IBM MQ for z/OS

Application Transparent Transport Layer Security (AT-TLS) proporciona soporte TLS para aplicaciones z/OS sin que dichas aplicaciones tengan que implementar el soporte TLS, o incluso tener en cuenta que se está utilizando TLS. AT-TLS solo está disponible en z/OS.

AT-TLS se puede utilizar con todas las versiones de IBM MQ for z/OS.

Antes de utilizar AT-TLS con IBM MQ for z/OS, asegúrese de que entiende el “Restricciones” en la página 456 implicado.

Para utilizar Application Transparent Transport Layer Security , defina sentencias de política que contengan un conjunto de reglas utilizadas por z/OS Communications Server para decidir qué conexiones TCP/IP tienen TLS habilitado de forma transparente.

IBM MQ for z/OS tiene su propia implementación TLS, que requiere que los canales tengan el parámetro SSLCIPH configurado con una CipherSpecsoportada.

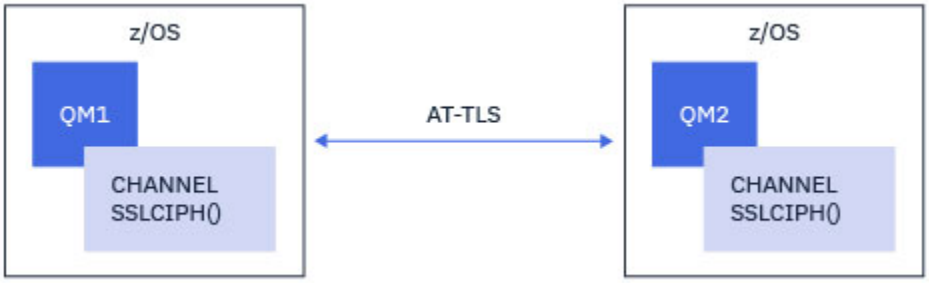
Al decidir habilitar TLS en un canal, el administrador de IBM MQ puede decidir utilizar AT-TLS o IBM MQ TLS. La decisión se suele tomar en función de si se utiliza AT-TLS para otro middleware o debido a implicaciones en el rendimiento. Para obtener una comparación básica del rendimiento de AT-TLS y IBM MQ TLS, consulte MP16: Capacity Planning and Tuning for IBM MQ for z/OS.

Escenarios

El uso de AT-TLS con IBM MQ está soportado en los escenarios siguientes:

Escenario 1

Entre dos gestores de colas de IBM MQ for z/OS en los que ambos lados del canal utilizan AT-TLS. Es decir, ninguno de los canales especifica el atributo SSLCIPH. Este enfoque se puede utilizar con cualquier canal de mensajes.



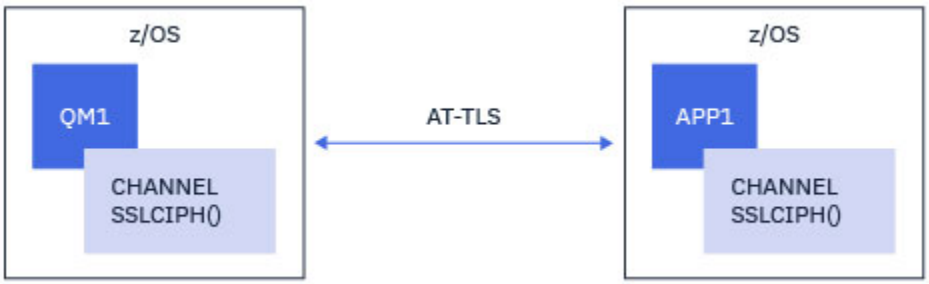
La implementación de este escenario consiste en definir dos políticas AT-TLS, una para cada lado del canal. Estas políticas son las mismas que las utilizadas con el Escenario 3 o el Escenario 4.

Por ejemplo, si el canal se estaba cambiando de utilizar un único CipherSpec a utilizar AT-TLS, el canal de salida utilizaría la política de "Configuración de AT-TLS en un canal de salida a un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec" en la página 457 y el canal de entrada utilizaría la política de "Configuración de AT-TLS en un canal de entrada desde un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec" en la página 466.

Si el canal se estaba cambiando de utilizar un alias CipherSpec a utilizar AT-TLS, el canal de salida utilizaría la política de "Configuración de AT-TLS en un canal de salida a un gestor de colas de IBM MQ for Multiplatforms utilizando el alias CipherSpecs" en la página 462 y el canal de entrada utilizaría la política de "Configuración de AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms utilizando un alias CipherSpec" en la página 470.

Escenario 2

Entre un gestor de colas de IBM MQ for z/OS y una aplicación cliente de IBM MQ Java que se ejecuta en z/OS donde ambos lados del canal utilizan AT-TLS. Es decir, ni el canal de conexión de servidor ni el canal de conexión de cliente especifican el atributo SSLCIPH.



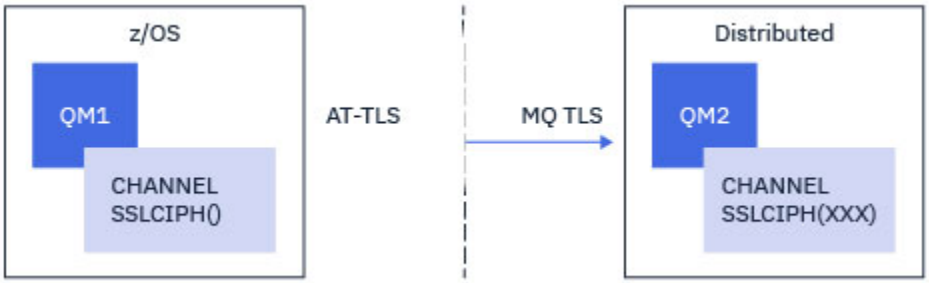
La implementación de este escenario consiste en definir dos políticas AT-TLS, una para cada lado del canal. Estas políticas son las mismas que las utilizadas con el Escenario 3 o el Escenario 4.

Por ejemplo, si el canal se estaba cambiando de utilizar un único CipherSpec a utilizar AT-TLS, el canal de conexión de cliente utilizaría la política de "Configuración de AT-TLS en un canal de salida a un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec" en la página 457 y el canal de conexión de servidor utilizaría la política de "Configuración de AT-TLS en un canal de entrada desde un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec" en la página 466.

Si el canal se estaba cambiando de utilizar un alias CipherSpec a utilizar AT-TLS, el canal de conexión de cliente utilizaría la política de "Configuración de AT-TLS en un canal de salida a un gestor de colas de IBM MQ for Multiplatforms utilizando el alias CipherSpecs" en la página 462 y el canal de conexión de servidor utilizaría la política de "Configuración de AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms utilizando un alias CipherSpec" en la página 470.

Escenario 3

entre unIBM MQ for z/OS gestor de colas y un gestor de colas ejecutándose enIBM MQ for Multiplatforms , donde elIBM MQ for z/OS El administrador de colas utiliza AT-TLS y elIBM MQ for Multiplatforms usos del administrador de colasIBM MQ TLS, especificando el atributo SSLCIPH con un único nombreCipherSpec . Esto se aplica a todos los tipos de canal de mensajes que no sean clúster emisor y clúster receptor.

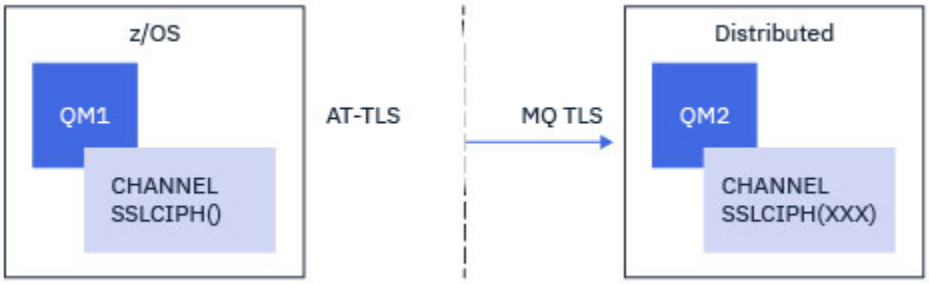


Consulte "Configuración de AT-TLS en un canal de salida a un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec" en la página 457 para ver un ejemplo de configuración AT-TLS para canales de salida desde el gestor de colas IBM MQ for z/OS al gestor de colas IBM MQ for Multiplatforms , y "Configuración de AT-TLS en un canal de entrada desde un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec" en la página 466 para ver un ejemplo de configuración AT-TLS para canales de entrada desde el gestor de colas IBM MQ for Multiplatforms al gestor de colas IBM MQ for z/OS .

Se puede utilizar la misma configuración AT-TLS cuando ambos gestores de colas están en z/OS, pero el gestor de colas de la derecha no se ha configurado para utilizar AT-TLS.

Escenario 4

Entre un gestor de colas de IBM MQ for z/OS y un gestor de colas que se ejecuta en IBM MQ for Multiplatforms, donde el gestor de colas de IBM MQ for z/OS utiliza AT-TLS y el gestor de colas de IBM MQ for Multiplatforms utiliza IBM MQ TLS, especificando el atributo SSLCIPH con un alias CipherSpec. Esto se aplica a todos los tipos de canal de mensajes que no sean clúster emisor y clúster receptor.

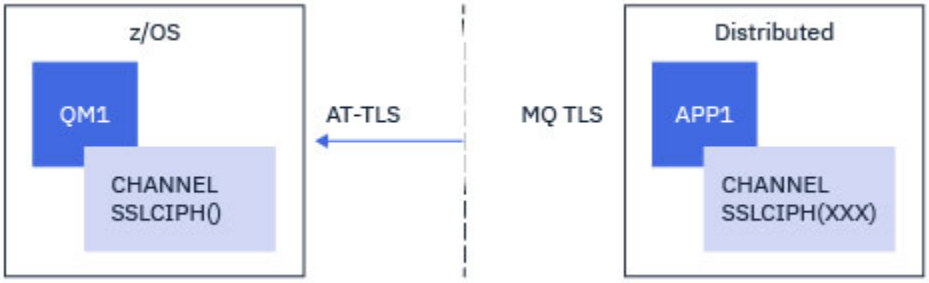


Consulte "Configuración de AT-TLS en un canal de salida a un gestor de colas de IBM MQ for Multiplatforms utilizando el alias CipherSpecs" en la página 462 para ver un ejemplo de configuración AT-TLS para canales de salida desde el gestor de colas IBM MQ for z/OS al gestor de colas IBM MQ for Multiplatforms , y "Configuración de AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms utilizando un alias CipherSpec" en la página 470, y "Configuración de AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms utilizando un alias CipherSpec" en la página 470 para ver un ejemplo de configuración AT-TLS para canales de entrada desde el gestor de colas IBM MQ for Multiplatforms al gestor de colas IBM MQ for z/OS .

Se puede utilizar la misma configuración AT-TLS cuando ambos gestores de colas están en z/OS, pero el gestor de colas de la derecha no se ha configurado para utilizar AT-TLS.

Escenario 5

Entre un gestor de colas de IBM MQ for z/OS y una aplicación cliente que se ejecuta en IBM MQ for Multiplatforms, donde el gestor de colas de IBM MQ for z/OS utiliza AT-TLS y la aplicación cliente utiliza IBM MQ TLS especificando el atributo SSLCIPH con una única, denominada CipherSpec.

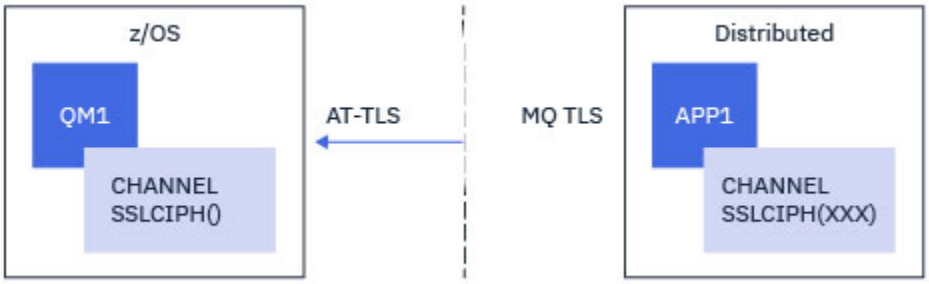


Este escenario requiere una única política AT-TLS que cumpla los mismos requisitos que los utilizados por un canal de mensajes de entrada; consulte "Configuración de AT-TLS en un canal de entrada desde un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec" en la página 466.

Se puede utilizar la misma configuración AT-TLS cuando la aplicación cliente es una aplicación Java y también se ejecuta en z/OS, pero no se ha configurado para utilizar AT-TLS.

Caso de ejemplo 6

Entre un gestor de colas de IBM MQ for z/OS y una aplicación cliente que se ejecuta en IBM MQ for Multiplatforms, donde el gestor de colas de IBM MQ for z/OS utiliza AT-TLS y la aplicación cliente utiliza IBM MQ TLS especificando el atributo SSLCIPH con un alias CipherSpec.



Este escenario requiere una única política AT-TLS que cumpla los mismos requisitos que los utilizados por un canal de mensajes de entrada; consulte "Configuración de AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms utilizando un alias CipherSpec" en la página 470.

Se puede utilizar la misma configuración AT-TLS cuando la aplicación cliente es una aplicación Java y también se ejecuta en z/OS, pero no se ha configurado para utilizar AT-TLS.

Restricciones

IBM MQ for z/OS no tiene en cuenta AT-TLS, por lo tanto, hay varias restricciones que se aplican con los escenarios anteriores:

- AT-TLS en combinación con IBM MQ TLS no funciona con los canales de clúster emisor y clúster receptor.
- Los gestores de colas de IBM MQ for z/OS no son conscientes de que están utilizando AT-TLS y no reciben ninguna información de certificado de su gestor de colas o cliente asociado. Por lo tanto, los atributos siguientes no tienen ningún efecto en el lado z/OS de un canal que utiliza AT-TLS:
  - Los atributos de canal SSLCAUTH y SSLPEER

  - Atributo del gestor de colas SSLRKEYC
  - Los atributos SSLPEERMAP de las reglas CHLAUTH
- El uso de la renegociación de claves secretas TLS requiere que ambos lados del canal utilicen IBM MQ TLS. Por lo tanto, un gestor de colas de IBM MQ for Multiplatforms , o cliente, no debe tener habilitada la renegociación de claves secretas TLS si se conecta a un gestor de colas de IBM MQ for z/OS utilizando AT-TLS.

  Para inhabilitar la renegociación de claves secretas TLS para un gestor de colas, establezca el parámetro SSLRKEYC del gestor de colas en `0`. Para un cliente, establezca el parámetro relevante en `0` en función del tipo de cliente. Para obtener detalles sobre cómo hacerlo, consulte "Restablecimiento de claves secretas SSL y TLS" en la página 475.

Sentencias de configuración AT-TLS

AT-TLS se configura utilizando un conjunto de sentencias. Los utilizados en los escenarios documentados en este tema son:

ReglaTTLS
: Especifica un conjunto de criterios para comparar una conexión TCP/IP con una configuración TLS. Esto a su vez hace referencia a los otros tipos de sentencia.

TTLSGroupAction
: Especifica si la `TTLSRule` de referencia está habilitada o no.

TTLSEnvironmentAction
: Especifica la configuración detallada para el `TTLSRule` de referencia y hace referencia a una serie de otras sentencias.

TTLSKeyringParms
: Hace referencia al conjunto de claves que utilizará AT-TLS.

TTLSCipherParms
: Define las suites de cifrado que se van a utilizar.

TTLSEnvironmentAdvancedParms
: Define qué protocolos TLS o SSL están habilitados.

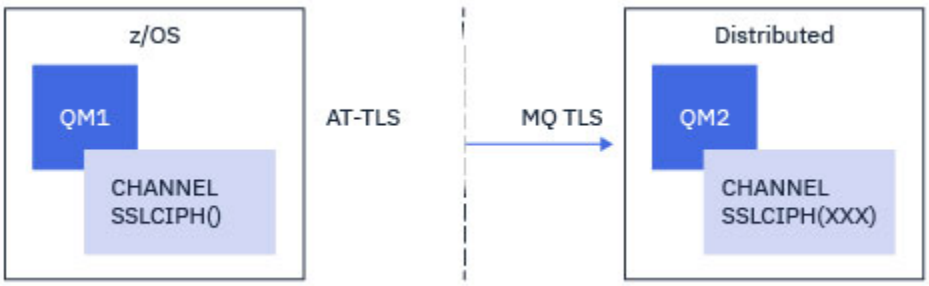
Atención: Existen otras sentencias de política AT-TLS con AT-TLS que no están documentadas aquí y que se pueden utilizar con IBM MQ en función de las necesidades. Sin embargo, IBM MQ sólo se ha probado con las políticas descritas en este tema.

z/OS *Configuración de AT-TLS en un canal de salida a un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec*

Cómo configurar AT-TLS en un canal de salida desde un gestor de colas de IBM MQ for z/OS a un gestor de colas de IBM MQ for Multiplatforms . En este caso, el canal del gestor de colas z/OS es un canal emisor que no tiene establecido el atributo SSLCIPH, y el canal del gestor de colas que no es dez/OS es un canal receptor con el atributo SSLCIPH establecido en un único, denominado CipherSpec.

Consulte "Configuración de AT-TLS en un canal de salida a un gestor de colas de IBM MQ for Multiplatforms utilizando el alias CipherSpecs" en la página 462 para ver un ejemplo de utilización de un alias CipherSpec.

En este ejemplo, un par de canales emisor-receptor existente, que utiliza TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec se va a ajustar para que el canal emisor utilice AT-TLS en lugar de IBM MQ TLS.



Se pueden utilizar otros protocolos TLS y CipherSpecs realizando ajustes menores en la configuración. Otros tipos de canal de mensajes, aparte de los canales de clúster emisor y de clúster receptor, se pueden utilizar sin ningún cambio en la configuración de AT-TLS.

Procedimiento

Paso 1: Detener el canal

Paso 2: Crear y aplicar una política AT-TLS

Debe crear las siguientes sentencias AT-TLS para este escenario:

1. Una sentencia TTLSRule para hacer coincidir las conexiones de salida del espacio de direcciones del iniciador de canal con la dirección IP y el número de puerto del canal receptor de destino. Estos valores deben coincidir con la información utilizada en el CONNAME del canal emisor. Aquí, se ha incluido un filtrado adicional para que coincida con un nombre de trabajo de 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
}
```

   La regla anterior coincide con las conexiones que van a la dirección IP 123.456.78.9 en el puerto 1414 del trabajo CSQ1CHIN .

   Las opciones de filtrado más avanzadas se describen en TTLSRule.

2. Una sentencia TTLSGroupAction que habilita la regla. `TTLSRule` hace referencia a `TTLSGroupAction` utilizando la propiedad **`TTLSGroupActionRef`** .

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Una sentencia TTLSEnvironmentAction asociada con `TTLSRule` mediante la propiedad **`TTLSEnvironmentActionRef`** . Un `TTLSEnvironmentAction` configura el entorno TLS y especifica qué conjunto de claves se debe utilizar.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. Una sentencia TTLSKeyringParms asociada con TTLSEnvironmentAction por la propiedad **TTLSKeyringParmsRef** y define el conjunto de claves utilizado por AT-TLS.

   El conjunto de claves debe contener certificados de confianza del gestor de colas remoto noz/OS . Este conjunto de claves se puede definir de la misma forma que un conjunto de claves utilizado por el iniciador de canal; consulte “Configuring your z/OS system to use TLS” en la página 260.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Una sentencia TTLSCipherParms asociada con TTLSEnvironmentAction mediante la propiedad **TTLSCipherParmsRef** .

   Esta sentencia debe contener un único nombre de suite de cifrado que debe ser el equivalente al nombre de IBM MQ CipherSpec utilizado en el canal receptor de destino.

   Nota: Los nombres de suite de cifrado AT-TLS no coinciden necesariamente con los nombres de IBM MQ CipherSpec . Sin embargo, es posible encontrar el nombre de la suite de cifrado AT-TLS que coincide con un nombre IBM MQ CipherSpec buscando el nombre IBM MQ CipherSpec en la tabla siguiente y haciendo referencia cruzada a la columna de código hexadecimal con la columna de caracteres expandida de la Tabla 2 en el tema de la sentencia TTLSCipherParms .

Tabla 83. CipherSpecs en z/OS desde IBM MQ for z/OS 9.2.0

CipherSpec	Protocolo	Código hexadecimal	Habilitado de forma predeterminada
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sí
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sí
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sí
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sí
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sí
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sí
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sí
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sí
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sí

Tabla 83. CipherSpecs en z/OS desde IBM MQ for z/OS 9.2.0 (continuación)

CipherSpec	Protocolo	Código hexadecimal	Habilitado de forma predeterminada
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sí
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sí
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sí
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sí
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. Una sentencia TTLSEnvironmentAdvancedParms está asociada con TTLSEnvironmentAction mediante la propiedad **TTLSEnvironmentAdvancedParmsRef** .

   Esta sentencia se puede utilizar para especificar qué protocolos SSL y TLS están habilitados. Con IBM MQ , solo debe habilitar el protocolo único que coincida con el nombre de suite de cifrado utilizado en la sentencia TTLSCipherParms .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

El conjunto completo de sentencias son las siguientes y se deben aplicar al agente de políticas:

```
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
}
```

Paso 3: Eliminar SSLCIPH del canal z/OS

Elimine la CipherSpec del canal z/OS utilizando el mandato siguiente:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Paso 4: Iniciar el canal

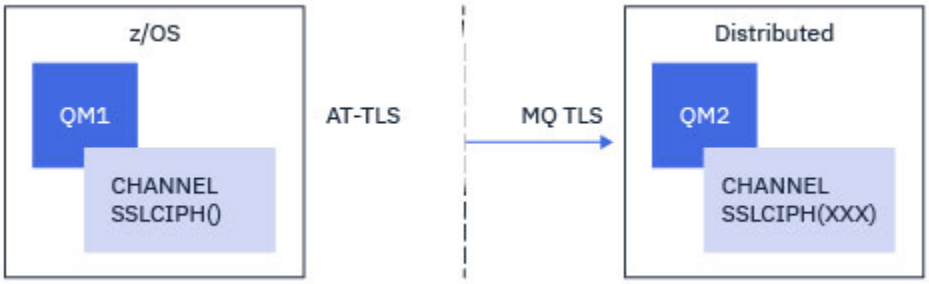
Una vez iniciado el canal, utilizará una combinación de AT-TLS y IBM MQ TLS.

Atención: Las sentencias AT-TLS anteriores son sólo una configuración mínima. Existen otras sentencias de política AT-TLS con AT-TLS que no están documentadas aquí y que se pueden utilizar con IBM MQ en función de las necesidades. Sin embargo, IBM MQ sólo se ha probado con las políticas descritas.

z/OS *Configuración de AT-TLS en un canal de salida a un gestor de colas de IBM MQ for Multiplatforms utilizando el alias CipherSpecs*

Cómo configurar AT-TLS en un canal de salida desde un gestor de colas de IBM MQ for z/OS a un gestor de colas de IBM MQ for Multiplatforms . En este caso, el canal del gestor de colas z/OS es un canal emisor que no tiene establecido el atributo SSLCIPH, y el canal del gestor de colas noz/OS es un canal receptor con el atributo SSLCIPH establecido en un alias CipherSpec

En este ejemplo, un par de canales emisor-receptor existente, que utiliza el alias ANY_TLS13 CipherSpec se va a ajustar para que el canal emisor utilice AT-TLS en lugar de IBM MQ TLS.



Se pueden utilizar otros protocolos TLS y CiperSpecs realizando ajustes menores en la configuración. Otros tipos de canal de mensajes, aparte de los canales de clúster emisor y de clúster receptor, se pueden utilizar sin ningún cambio en la configuración de AT-TLS.

Procedimiento

Paso 1: Detener el canal

Paso 2: Crear y aplicar una política AT-TLS

Debe crear las siguientes sentencias AT-TLS para este escenario:

1. Una sentencia TTLSRule para hacer coincidir las conexiones de salida del espacio de direcciones del iniciador de canal con la dirección IP y el número de puerto del canal receptor de destino. Estos valores deben coincidir con la información utilizada en el CONNAME del canal emisor. Aquí, se ha incluido un filtrado adicional para que coincida con un nombre de trabajo de 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
}
```

   La regla anterior coincide con las conexiones que van a la dirección IP 123.456.78.9 en el puerto 1414 del trabajo CSQ1CHIN .

   Las opciones de filtrado más avanzadas se describen en TTLSRule.

2. Una sentencia TTLSGroupAction que habilita la regla. TTLSRule hace referencia a TTLSGroupAction utilizando la propiedad **TTLSGroupActionRef** .

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Una sentencia TTLSEnvironmentAction asociada con TTLSRule mediante la propiedad **TTLSEnvironmentActionRef** . Un TTLSEnvironmentAction configura el entorno TLS y especifica qué conjunto de claves se debe utilizar.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. Una sentencia TTLSKeyringParms asociada con TTLSEnvironmentAction por la propiedad **TTLSKeyringParmsRef** y define el conjunto de claves utilizado por AT-TLS.

   El conjunto de claves debe contener certificados de confianza del gestor de colas remoto noz/OS . Este conjunto de claves se puede definir de la misma forma que un conjunto de claves utilizado por el iniciador de canal; consulte “Configuring your z/OS system to use TLS” en la página 260.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Una sentencia TTLSCipherParms asociada con TTLSEnvironmentAction mediante la propiedad **TTLSCipherParmsRef** .

   Esta sentencia debe contener uno o más nombres de suite de cifrado, al menos uno de los cuales debe ser compatible con el conjunto de CipherSpecs que implica el alias CipherSpec utilizado en el canal receptor de destino.

   Nota: Los nombres de suite de cifrado AT-TLS no coinciden necesariamente con los nombres de IBM MQ CipherSpec . Sin embargo, es posible encontrar el nombre de la suite de cifrado AT-TLS que coincida con un nombre IBM MQ CipherSpec buscando el nombre IBM MQ CipherSpec en la tabla siguiente y haciendo una referencia cruzada a la columna de código hexadecimal con la columna de caracteres expandida de la Tabla 2 en el tema TTLSCipherParms .

Tabla 84. CipherSpecs en z/OS desde IBM MQ for z/OS 9.2.0

CipherSpec	Protocolo	Código hexadecimal	Habilitado de forma predeterminada
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sí
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sí
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sí
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sí
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sí
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sí

Tabla 84. CipherSpecs en z/OS desde IBM MQ for z/OS 9.2.0 (continuación)

CipherSpec	Protocolo	Código hexadecimal	Habilitado de forma predeterminada
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sí
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sí
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sí
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sí
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sí
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sí
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sí
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
}
```

Atención: Si tanto el gestor de colas como la política AT-TLS dan soporte a TLS 1.3, sólo los alias CipherSpecs que contienen al menos una TLS 1.3 CipherSpec permiten que se inicie el canal. Por ejemplo, el uso de ANY_TLS12 hace que el canal no se inicie, incluso si TTLSCipherParms contiene TLS 1.2 CipherSpecs, pero el uso de ANY_TLS12_OR_HIGHER o ANY_TLS13 permite que se inicie el canal. Consulte "Relación entre los valores de alias CipherSpec" en la página 449 para obtener una explicación.

6. Una sentencia TTLSEnvironmentAdvancedParms está asociada con TTLSEnvironmentAction mediante la propiedad **TTLSEnvironmentAdvancedParmsRef** .

   Esta sentencia se puede utilizar para especificar qué protocolos SSL y TLS están habilitados y deben ser coherentes con las suites de cifrado de la sentencia TTLSCipherParms .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

El conjunto completo de sentencias son las siguientes y se deben aplicar al agente de políticas:

```
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
}
```

Paso 3: Eliminar SSLCIPH del canal z/OS

Elimine la CipherSpec del canal z/OS utilizando el mandato siguiente:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Paso 4: Iniciar el canal

Una vez iniciado el canal, utilizará una combinación de AT-TLS y IBM MQ TLS.

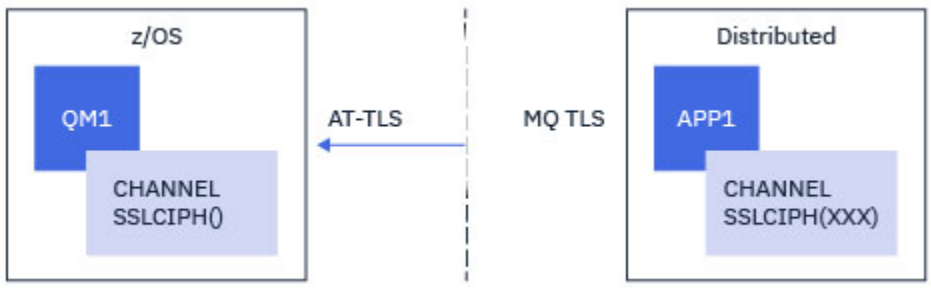
Atención: Las sentencias AT-TLS anteriores son sólo una configuración mínima. Existen otras sentencias de política AT-TLS con AT-TLS que no están documentadas aquí y que se pueden utilizar con IBM MQ en función de las necesidades. Sin embargo, IBM MQ sólo se ha probado con las políticas descritas.

z/OS *Configuración de AT-TLS en un canal de entrada desde un gestor de colas IBM MQ for Multiplatforms utilizando una única, denominada CipherSpec*

Cómo configurar AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms a un gestor de colas de IBM MQ for z/OS . En este caso, el canal del gestor de colas z/OS es un canal receptor que no tiene establecido el atributo SSLCIPH, y el canal del gestor de colas noz/OS es un canal emisor con el atributo SSLCIPH establecido en un único, denominado CipherSpec.

Consulte “Configuración de AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms utilizando un alias CipherSpec” en la página 470 para ver un ejemplo de utilización de un alias CipherSpec.

En este ejemplo, un par de canales emisor-receptor existente, que utiliza TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec se va a ajustar para que el canal receptor utilice AT-TLS en lugar de IBM MQ TLS.



Se pueden utilizar otros protocolos TLS y CipherSpecs realizando ajustes menores en la configuración. Otros tipos de canal de mensajes, aparte de los canales de clúster emisor y de clúster receptor, se pueden utilizar sin ningún cambio en la configuración de AT-TLS.

Procedimiento

Paso 1: Detener el canal

Paso 2: Crear y aplicar una política AT-TLS

Debe crear las siguientes sentencias AT-TLS para este escenario:

1. Una sentencia TTLSRule para hacer coincidir las conexiones de entrada con el espacio de direcciones del iniciador de canal desde la dirección IP del canal emisor. Aquí, se ha incluido un filtrado adicional para que coincida con un nombre de trabajo de 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
}
```

La regla anterior coincide con las conexiones que entran en el trabajo CSQ1CHIN en el puerto local 1414 desde la dirección IP remota 123.456.78.9.

Las opciones de filtrado más avanzadas se describen en TTLSRule.

2. Una sentencia TTLSGroupAction que habilita la regla. TTLSRule hace referencia a TTLSGroupAction utilizando la propiedad **TTLSGroupActionRef** .

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Una sentencia TTLSEnvironmentAction se asocia con TTLSRule mediante la propiedad **TTLSEnvironmentActionRef** . Un TTLSEnvironmentAction configura el entorno TLS y especifica qué conjunto de claves se debe utilizar.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS proporciona la posibilidad de proporcionar autenticación mutua, que es el equivalente a utilizar el atributo de canal SSLCAUTH. Esto se realiza teniendo una sentencia TTLSEnvironmentAction con un valor **HandshakeRole** de *ServerWithClientAuth* para la sentencia TTLSEnvironmentAction de entrada.

4. Una sentencia TTLSKeyringParms se asocia con TTLSEnvironmentAction mediante la propiedad **TTLSKeyringParmsRef** y define el conjunto de claves utilizado por AT-TLS.

   El conjunto de claves debe contener certificados de confianza del gestor de colas remoto noz/OS . Este conjunto de claves se puede definir de la misma forma que un conjunto de claves utilizado por el iniciador de canal; consulte “Configuring your z/OS system to use TLS” en la página 260.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Una sentencia TTLSCipherParms asociada con TTLSEnvironmentAction mediante la propiedad **TTLSCipherParmsRef** .

   Esta sentencia debe contener un único nombre de suite de cifrado que debe ser el equivalente al nombre de IBM MQ CipherSpec utilizado en el canal emisor remoto.

   Nota: Los nombres de suite de cifrado AT-TLS no coinciden necesariamente con los nombres de IBM MQ CipherSpec . Sin embargo, es posible encontrar el nombre de la suite de cifrado AT-TLS que coincide con un nombre IBM MQ CipherSpec buscando el nombre IBM MQ CipherSpec en la tabla siguiente y haciendo referencia cruzada a la columna de código hexadecimal con la columna de caracteres expandida de la Tabla 2 en el tema de la sentencia TTLSCipherParms .

Tabla 85. CipherSpecs en z/OS desde IBM MQ for z/OS 9.2.0			
CipherSpec	**Protocolo**	**Código hexadecimal**	**Habilitado de forma predeterminada**
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sí
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sí
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sí
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sí
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sí
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sí
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sí
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sí
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sí
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sí
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sí
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sí
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sí
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

Tabla 85. CipherSpecs en z/OS desde IBM MQ for z/OS 9.2.0 (continuación)			
CipherSpec	**Protocolo**	**Código hexadecimal**	**Habilitado de forma predeterminada**
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. Una sentencia TTLSEnvironmentAdvancedParms está asociada con TTLSEnvironmentAction mediante la propiedad **TTLSEnvironmentAdvancedParmsRef** .

   Esta sentencia se puede utilizar para especificar qué protocolos SSL y TLS están habilitados. Con IBM MQ , solo debe habilitar el protocolo único que coincida con el nombre de suite de cifrado utilizado en la sentencia TTLSCipherParms .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

El conjunto completo de sentencias son las siguientes y se deben aplicar al agente de políticas:

```
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
}
```

Paso 3: Eliminar SSLCIPH del canal z/OS

Elimine la CipherSpec del canal z/OS utilizando el mandato siguiente:

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

Paso 4: Iniciar el canal

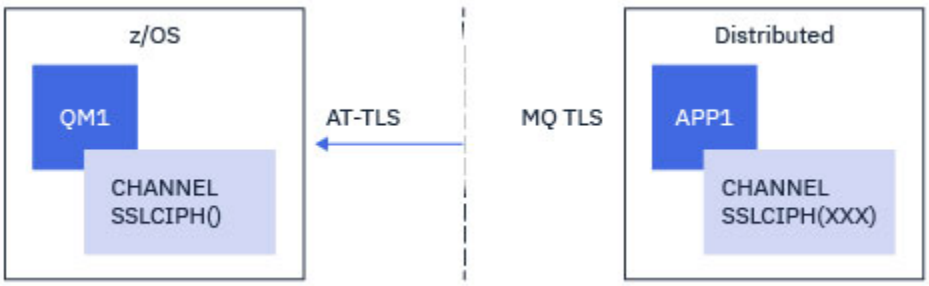
Una vez iniciado el canal, utilizará una combinación de AT-TLS y IBM MQ TLS.

Atención: Las sentencias AT-TLS anteriores son sólo una configuración mínima. Existen otras sentencias de política AT-TLS con AT-TLS que no están documentadas aquí y que se pueden utilizar con IBM MQ en función de las necesidades. Sin embargo, IBM MQ sólo se ha probado con las políticas descritas.

z/OS *Configuración de AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms utilizando un alias CipherSpec*

Cómo configurar AT-TLS en un canal de entrada desde un gestor de colas de IBM MQ for Multiplatforms a un gestor de colas de IBM MQ for z/OS . En este caso, el canal del gestor de colas z/OS es un canal receptor que no tiene establecido el atributo SSLCIPH, y el canal del gestor de colas que no es dez/OS es un canal emisor con el atributo SSLCIPH establecido en un alias CipherSpec.

En este ejemplo, un par de canales emisor-receptor existente, que utiliza cualquier TLS 1.3 CipherSpec se va a ajustar para que el canal receptor utilice AT-TLS en lugar de IBM MQ TLS.



Se pueden utilizar otros protocolos TLS y CiperSpecs realizando ajustes menores en la configuración. Otros tipos de canal de mensajes, aparte de los canales de clúster emisor y de clúster receptor, se pueden utilizar sin ningún cambio en la configuración de AT-TLS.

Procedimiento

Paso 1: Detener el canal

Paso 2: Crear y aplicar una política AT-TLS

Debe crear las siguientes sentencias AT-TLS para este escenario:

1. Una sentencia TTLSRule para hacer coincidir las conexiones de entrada con el espacio de direcciones del iniciador de canal desde la dirección IP del canal emisor. Aquí, se ha incluido un filtrado adicional para que coincida con un nombre de trabajo de 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
}
```

   La regla anterior coincide con las conexiones que entran en el trabajo CSQ1CHIN en el puerto local 1414 desde la dirección IP remota 123.456.78.9.

   Las opciones de filtrado más avanzadas se describen en TTLSRule.

2. Una sentencia TTLSGroupAction que habilita la regla. `TTLSRule` hace referencia a `TTLSGroupAction` utilizando la propiedad **`TTLSGroupActionRef`** .

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. Una sentencia TTLSEnvironmentAction se asocia con `TTLSRule` mediante la propiedad **`TTLSEnvironmentActionRef`** . Un `TTLSEnvironmentAction` configura el entorno TLS y especifica qué conjunto de claves se debe utilizar.

```
TTLSEnvironmentAction                CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                      SERVER
  TTLSKeyringParmsRef                CSQ1-KEYRING
  TTLSCipherParmsRef                 CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef    CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS proporciona la posibilidad de proporcionar autenticación mutua, que es el equivalente a utilizar el atributo de canal SSLCAUTH. Esto se realiza teniendo una sentencia TTLSEnvironmentAction con un valor **HandshakeRole** de *ServerWithClientAuth* para la sentencia TTLSEnvironmentAction de entrada.

4. Una sentencia TTLSKeyringParms se asocia con TTLSEnvironmentAction mediante la propiedad **TTLSKeyringParmsRef** y define el conjunto de claves utilizado por AT-TLS.

   El conjunto de claves debe contener certificados de confianza del gestor de colas remoto noz/OS . Este conjunto de claves se puede definir de la misma forma que un conjunto de claves utilizado por el iniciador de canal; consulte “Configuring your z/OS system to use TLS” en la página 260.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. Una sentencia TTLSCipherParms asociada con TTLSEnvironmentAction mediante la propiedad **TTLSCipherParmsRef** .

   Esta sentencia debe contener al menos un nombre de suite de cifrado que se incluya en el alias CipherSpec establecido en el canal emisor remoto.

   Nota: Los nombres de suite de cifrado AT-TLS no coinciden necesariamente con los nombres de IBM MQ CipherSpec . Sin embargo, es posible encontrar el nombre de la suite de cifrado AT-TLS que coincide con un nombre IBM MQ CipherSpec buscando el nombre IBM MQ CipherSpec en la tabla siguiente y haciendo referencia cruzada a la columna de código hexadecimal con la columna de caracteres expandida de la Tabla 2 en el tema de la sentencia TTLSCipherParms .

Tabla 86. CipherSpecs en z/OS desde IBM MQ for z/OS 9.2.0

CipherSpec	Protocolo	Código hexadecimal	Habilitado de forma predeterminada
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Sí
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Sí
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Sí
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Sí
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Sí
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Sí
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Sí
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Sí
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Sí
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Sí

Tabla 86. CipherSpecs en z/OS desde IBM MQ for z/OS 9.2.0 (continuación)			
CipherSpec	**Protocolo**	**Código hexadecimal**	**Habilitado de forma predeterminada**
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Sí
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Sí
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Sí
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
}
```

Atención: Si tanto el gestor de colas como la política AT-TLS dan soporte a TLS 1.3, sólo los alias CipherSpecs que contienen al menos una TLS 1.3 CipherSpec permiten que se inicie el canal. Por ejemplo, el uso de ANY_TLS12 hace que el canal no se inicie, incluso si TTLSCipherParms contiene TLS 1.2 CipherSpecs, pero el uso de ANY_TLS12_OR_HIGHER o ANY_TLS13 permite que se inicie el canal. Consulte "Relación entre los valores de alias CipherSpec" en la página 449 para obtener una explicación.

6. Una sentencia TTLSEnvironmentAdvancedParms está asociada con TTLSEnvironmentAction mediante la propiedad **TTLSEnvironmentAdvancedParmsRef** .

Esta sentencia se puede utilizar para especificar qué protocolos SSL y TLS están habilitados y deben ser coherentes con las suites de cifrado de la sentencia `TTLSCipherParms` .

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

El conjunto completo de sentencias son las siguientes y se deben aplicar al agente de políticas:

```
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
}
```

Paso 3: Eliminar SSLCIPH del canal z/OS

Elimine la CipherSpec del canal z/OS utilizando el mandato siguiente:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Paso 4: Iniciar el canal

Una vez iniciado el canal, utilizará una combinación de AT-TLS y IBM MQ TLS.

Atención: Las sentencias AT-TLS anteriores son sólo una configuración mínima. Existen otras sentencias de política AT-TLS con AT-TLS que no están documentadas aquí y que se pueden utilizar con IBM MQ en función de las necesidades. Sin embargo, IBM MQ sólo se ha probado con las políticas descritas.

Restablecimiento de claves secretas SSL y TLS

IBM MQ da soporte al restablecimiento de claves secretas en gestores de colas y clientes.

Las claves secretas se restablecen cuando un número especificado de bytes de datos cifrados han fluido a través del canal. Si las pulsaciones de canal están habilitadas, la clave secreta se restablece antes de que se envíen o reciban datos después de una pulsación de canal.

El valor de restablecimiento de clave siempre se establece inicializando el lado del canal de IBM MQ.

Gestor de colas

Para un gestor de colas, utilice el mandato **ALTER QMGR** con el parámetro **SSLRKEYC** para establecer los valores utilizados durante la renegociación de claves.

IBM i En IBM i, utilice **CHGMQM** con el parámetro **SSLRSTCNT** .

Cliente MQI

De forma predeterminada, los clientes MQI no renegocian la clave secreta. Puede hacer que un cliente MQI renegocie la clave de tres formas. En la lista siguiente, los métodos se muestran en orden de prioridad. Si especifica varios valores, se utiliza el valor de prioridad más alto.

1. Utilizando el campo KeyResetde conjunto de claves de la estructura MQSCO en una llamada MQCONNX.
2. Utilizando la variable de entorno **MQSSLRESET**.
3. Estableciendo el atributo **SSLKeyResetCount** en la stanza SSL del archivo de configuración de cliente.

Estas variables se pueden establecer en un número entero comprendido entre 0 y 999.999.999, representando el número de bytes no cifrados enviados y recibidos en una conversación TLS antes de que la clave secreta TLS se vuelva a negociar. Especificar un valor 0 indica que las claves secretas TLS no se renegocian nunca. Si especifica una cuenta de restablecimiento de clave secreta TLS entre 1 byte y 32 KB, los canales TLS utilizarán una cuenta de restablecimiento de clave secreta de 32 KB. De esta forma, se evitan restablecimientos de clave excesivos que se producirían para valores de restablecimiento de claves secretas TLS pequeñas.

Si se especifica un valor superior a 0 y las pulsaciones del canal están habilitadas para el canal, la clave secreta también se vuelve a negociar antes de que se envíen o se reciban datos de mensaje tras una pulsación del canal.

El número de bytes hasta la siguiente negociación de la clave secreta se restablece después de cada negociación satisfactoria.

Java

Para IBM MQ classes for Java, una aplicación puede restablecer la clave secreta en cualquiera de las maneras siguientes:

- Estableciendo el campo sslResetCount en la clase MQEnvironment.
- Estableciendo la propiedad de entorno MQC.SSL_RESET_COUNT_PROPERTY en un objeto Hashtable. A continuación, la aplicación asigna la tabla hash al campo `properties` en la clase MQEnvironment o pasa la tabla hash a un objeto MQQueueManager de su constructor.

Si la aplicación utiliza más de uno de estos métodos, se aplican las reglas de prioridad habituales. Consulte Clase com.ibm.mq.MQEnvironment para las reglas de prioridad.

El valor del campo sslResetCount o de la propiedad de entorno MQC.SSL_RESET_COUNT_PROPERTY representa el número total de bytes enviados y recibidos por el código de cliente IBM MQ classes for Java antes de que se renegocie la clave secreta. El número de bytes enviados es el número antes del cifrado y el número de bytes recibidos es el número después del cifrado. El número de bytes incluye también la información de control enviada y recibida por el cliente IBM MQ classes for Java.

Si la cuenta de restablecimiento es cero, que es el valor predeterminado, la clave secreta nunca se renegocia. La cuenta de restablecimiento se ignora si no se especifica ninguna CipherSuite.

JMS

Para IBM MQ classes for JMS, la propiedad SSLRESETCOUNT representa el número total de bytes enviados y recibidos por una conexión antes de renegociar la clave secreta que se utiliza para el cifrado. El número de bytes enviados es el número antes del cifrado y el número de bytes recibidos es el número después del cifrado. El número de bytes también incluye información de control enviada y recibida por IBM MQ classes for JMS. Por ejemplo, para configurar un objeto ConnectionFactory que se puede utilizar para crear una conexión a través de un canal MQI habilitado para TLS con una clave secreta que se renegocia después de que hayan fluido 4 MB de datos, emita el mandato siguiente en JMSAdmin:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

Si el valor de SSLRESETCOUNT es cero, que es el valor predeterminado, la clave secreta nunca se renegocia. La propiedad SSLRESETCOUNT se ignora si SSLCIPHERSUITE no está establecido.

.NET

Para clientes no gestionados de .NET , la propiedad de entero **SSLKeyResetCount** indica el número de bytes no cifrados enviados y recibidos dentro de una conversación TLS antes de que se renegocie la clave secreta. Para obtener más información sobre el uso de propiedades de objeto en IBM MQ classes for .NET, consulte Obtención y establecimiento de valores de atributo.

En los clientes gestionados de .NET, la clase SSLStream no soporta restablecimiento/negociación de clave secreta. Sin embargo, para ser coherente con otros clientes IBM MQ , el cliente IBM MQ gestionado .NET permite que las aplicaciones establezcan **SSLKeyResetCount**. Para obtener más información, consulte Restablecimiento o renegociación de una clave secreta.

XMS .NET

En clientes no gestionados de .NET XMS, consulte Conexiones seguras con un gestor de colas de IBM MQ.

Referencia relacionada
ALTER QMGR
DISPLAYQMGR
Cambiar gestor de colas de mensajes (CHGMQM)
Visualizar gestor de colas de mensajes (DSPMQM)

Implementación de confidencialidad en programas de salida de usuario

Implementación de confidencialidad en salidas de seguridad

Las salidas de seguridad pueden jugar un papel en el servicio de confidencialidad, generando y distribuyendo la clave simétrica para cifrar y descifrar los datos que fluyen en el canal. Una técnica común para hacerlo utiliza la tecnología PKI.

Una salida de seguridad genera un valor de datos aleatorio, lo cifra con la clave pública del gestor de colas o usuario al que representa la salida de seguridad del asociado y envía los datos cifrados a su asociado en un mensaje de seguridad. La salida de seguridad del asociado descifra el valor de datos aleatorio con la clave privada de gestor de claves o usuario al que representa. Ahora cada salida de seguridad

puede utilizar el valor de datos aleatorio para obtener la clave simétrica, independientemente de la otra, utilizando un algoritmo que ambas conocen. Como alternativa, pueden utilizar el valor de datos aleatorio como clave.

Si la primera salida de seguridad aún no ha autenticado a su asociado, el siguiente mensaje de seguridad que envía el asociado puede contener un valor esperado cifrado con la clave simétrica. Ahora la primera salida de seguridad puede autenticar a su asociado comprobando que la salida de seguridad del asociado ha podido cifrar correctamente el valor esperado.

Las salidas de seguridad también pueden utilizar esta oportunidad para acordar el algoritmo para cifrar y descifrar los datos que fluyen en el canal, en el caso de que se pueda utilizar más de un algoritmo.

Implementación de confidencialidad en salidas de mensajes

Una salida de mensajes del extremo emisor de un canal puede cifrar los datos de aplicación en un mensaje y otra salida de mensajes en el extremo receptor del canal puede descifrar los datos. Por razones de rendimiento, para esta finalidad se utiliza normalmente un algoritmo de clave simétrica. Para obtener más información sobre cómo se puede generar y distribuir la clave simétrica, consulte el apartado “Implementación de confidencialidad en programas de salida de usuario” en la página 476.

Las cabeceras de un mensaje, como la cabecera de la cola de transmisión MQXQH, que incluye el descriptor de mensaje incorporado, no se pueden cifrar mediante una salida de mensajes. Esto se debe a que la conversión de datos de las cabeceras de mensajes se lleva a cabo después de que se llame a la salida de mensajes en el extremo emisor o antes de que se llame a la salida de mensajes en el extremo receptor. Si las cabeceras están cifradas, la conversión de datos da error y el canal se detiene.

Implementación de confidencialidad en salidas de emisión y recepción

Las salidas de emisión y recepción se pueden utilizar para cifrar y descifrar los datos que fluyen en un canal. Resultan más adecuadas que las salidas de mensajes para proporcionar este servicio por los siguientes motivos:

- En un canal de mensajes, las cabeceras de mensajes se pueden cifrar, al igual que los datos de aplicación de los mensajes.
- Las salidas de emisión y recepción se pueden utilizar tanto en canales MQI como en canales de mensajes. Los parámetros de las llamadas MQI pueden contener datos que dependan de la aplicación que se tengan que proteger mientras fluyen en un canal MQI. Por lo tanto, puede utilizar las mismas salidas de emisión y recepción en ambos tipos de canales.

Implementación de confidencialidad en la salida de API y la salida cruzada de API

Una salida de API o salida cruzada de API puede cifrar los datos de aplicación de un mensaje cuando la aplicación emisora transfiere el mensaje y una segunda salida puede descifrarlos cuando la aplicación receptora recupera el mensaje. Por razones de rendimiento, para esta finalidad se utiliza normalmente un algoritmo de clave simétrica. No obstante, a nivel de aplicación, en el que muchos usuarios se pueden estar enviando mensajes, el problema es garantizar que sólo el receptor al que va destinado un mensaje pueda descifrar el mensaje. Una solución es utilizar una clave simétrica diferente para cada par de usuarios que se envían mensajes entre sí. Pero administrar esta solución puede resultar difícil y requerir mucho tiempo, sobretodo si los usuarios pertenecen a organizaciones diferentes. Un método estándar de resolver este problema es el que se conoce como *sobre digital* y utiliza la tecnología PKI.

Cuando una aplicación transfiere un mensaje a una cola, una salida de API o salida cruzada de API genera una clave simétrica aleatoria y utiliza la clave para cifrar los datos de aplicación incluidos en el mensaje. La salida cifra la clave simétrica con la clave pública del receptor al que va destinado. A continuación, sustituye los datos de aplicación del mensaje por los datos de aplicación cifrados y la clave simétrica cifrada. De este modo, solamente el receptor al que va destinado puede descifrar la clave simétrica y, por lo tanto, los datos de aplicación. Si un mensaje cifrado va destinado a más de un receptor, la salida puede cifrar una copia de la clave simétrica para cada receptor al que va destinado.

Si se dispone de algoritmos diferentes para cifrar y descifrar los datos de aplicación, la salida puede incluir el nombre del algoritmo que ha utilizado.

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 478
- Archive log data sets; see note “2” on page 478
- Page sets; see note “1” on page 478
- BSDS; see note “2” on page 478
- CSQINP* data sets; see note “2” on page 478
- SMDS; see note “1” on page 478

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 479 to "6" on page 479 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 479
2. "Configuring data set encryption for the log data sets" on page 480

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 480.

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 480

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 479

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 480 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 479
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 479.
   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 483.

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 484 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 483.

   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 483 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 483 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).

Integridad de datos de mensajes

Para mantener la integridad de los datos, puede utilizar varios tipos de programas de salida de usuario para proporcionar los resúmenes de mensajes o firmas digitales para los mensajes.

Integridad de datos

Implementación de la integridad de datos en los mensajes

Cuando se utiliza TLS, la opción de CipherSpec determina el nivel de integridad de datos en la empresa. Si utiliza IBM MQ Advanced Message Service (AMS), puede especificar la integridad de un mensaje exclusivo.

Implementación de la integridad de datos en salidas de mensajes

Un mensaje se puede firmar digitalmente mediante una salida de mensajes en el extremo emisor de un canal. Luego se puede comprobar la firma digital mediante una salida de mensajes en el extremo receptor de un canal para detectar si se ha modificado deliberadamente.

Se puede proporcionar cierta protección utilizando un resumen de mensaje en lugar de una firma digital. Un resumen de mensaje puede resultar eficaz frente a una manipulación casual o indiscriminada, pero no evita que una persona más informada modifique o sustituya el mensaje y genere para el mismo un resumen completamente nuevo. Esto resulta especialmente cierto si el algoritmo utilizado para generar el resumen de mensaje es muy conocido.

Implementación de la integridad de datos en salidas de emisión y recepción

En un canal de mensajes, las salidas de mensajes resultan más adecuadas para proporcionar este servicio porque una salida de mensajes tiene acceso al mensaje completo. En un canal MQI, los parámetros de llamadas MQI pueden contener datos de aplicación que se tengan que proteger, y sólo las salidas de emisión y recepción pueden proporcionar esta protección.

Implementación de la integridad de los datos en la salida de API o la salida cruzada de API

Una salida de API o salida cruzada de API puede firmar digitalmente un mensaje cuando la aplicación emisora transfiere el mensaje. La firma digital puede comprobarse mediante una segunda salida cuando la aplicación receptor recupera el mensaje para detectar si el mensaje ha sido modificado de forma deliberada.

Se puede proporcionar cierta protección utilizando un resumen de mensaje en lugar de una firma digital. Un resumen de mensaje puede resultar eficaz frente a una manipulación casual o indiscriminada, pero no evita que una persona más informada modifique o sustituya el mensaje y genere para el mismo un resumen completamente nuevo. Esto resulta especialmente cierto si el algoritmo utilizado para generar el resumen de mensaje es muy conocido.

Información adicional

Consulte la sección “Habilitación de CipherSpecs” en la página 428 para obtener más información sobre cómo garantizar la integridad de los datos.

Tareas relacionadas

Conexión de dos gestores de colas utilizando TLS

Conexión de un cliente a un gestor de colas de forma segura

Auditoría

Puede comprobar las intrusiones de seguridad, o intentos de intrusión, mediante mensajes de sucesos. También puede comprobar la seguridad del sistema utilizando IBM MQ Explorer.

Para detectar intentos de realizar acciones no autorizadas a tales como conectarse a un gestor de colas o transferir un mensaje a una cola, examine los mensajes de suceso generados por los gestores de colas, particularmente los mensajes de sucesos de autorización. Si desea más información sobre los mensajes de suceso del gestor de colas, consulte Sucesos del gestor de colas y si desea más información sobre la supervisión de sucesos en general, consulte Supervisión de sucesos.

Mantenimiento de la seguridad de los clústeres

Autorice o impida que los gestores de colas unan clústeres o coloquen mensajes en colas de clúster. Obligue a un gestor de colas a abandonar un clúster. Tenga en cuenta algunas consideraciones adicionales al configurar TLS para los clústeres.

Impedir que los gestores de colas no autorizados envíen mensajes

Impida que los gestores de colas no autorizados envíen mensajes a su gestor de colas utilizando una salida de seguridad de canal.

Antes de empezar

La agrupación en clúster no tiene ningún efecto en la manera en que funcionan las salidas de seguridad. Puede restringir el acceso a un gestor de colas igual que lo haría en un entorno de gestión de colas distribuidas.

Acerca de esta tarea

Impida que gestores de colas seleccionados envíen mensajes a su gestor de colas:

Procedimiento

1. Defina un programa de salida de seguridad de canal en la definición de canal CLUSRCVR.
2. Escriba un programa que autentique a los gestores de colas que intentan enviar mensajes en su canal de clúster receptor y que les deniegue el acceso si no están autorizados.

Qué hacer a continuación

Los programas de salida de seguridad de canal se invocan en la iniciación y la terminación del MCA.

Cómo hacer que los gestores de colas sin autorización pongan mensajes en sus colas

Utilice el atributo de canal Autorización de transferencia en el canal de clúster receptor para impedir que los gestores de colas no autorizados transfieran mensajes a sus colas. Autorice un gestor de colas remoto comprobando el ID de usuario en el mensaje utilizando RACF en z/OS, o el OAM en Multiplatforms.

Acerca de esta tarea

Utilice los recursos de seguridad de una plataforma y el mecanismo de control de acceso de IBM MQ para controlar el acceso a las colas.

Procedimiento

1. Para impedir que ciertos gestores de colas transfieran mensajes a una cola, utilice los recursos de seguridad disponibles en su plataforma.

   Por ejemplo:

   - z/OS RACF u otros gestores de seguridad externos en IBM MQ for z/OS
   - Multi El gestor de autorizaciones sobre objetos (OAM) en otras multiplataformas.
2. Utilice el atributo de autorización de transferencia, PUTAUT, en la definición de canal CLUSRCVR.

   El atributo PUTAUT le permite especificar qué identificadores de usuario se van a utilizar para establecer la autorización para transferir un mensaje a una cola.

   Las opciones del atributo PUTAUT son:

   DEF
   Utilice el ID de usuario predeterminado.

   z/OS En z/OS, la comprobación puede implicar el uso tanto del ID de usuario recibido de la red como del derivado de MCAUSER.

   CTX
   Utilizar el ID de usuario en la información de contexto asociada al mensaje.

   z/OS En z/OS, la comprobación puede implicar el uso del ID de usuario recibido de la red, del derivado de MCAUSER, o de ambos. Utilice esta opción si el enlace es fiable y está autenticado.

   z/OS **ONLYMCA (solamente z/OS)**
   Como en DEF, pero no se utiliza ningún ID de usuario recibido de la red. Utilice esta opción si el enlace no es fiable. Permita en él sólo un conjunto específico de acciones, que se definen para MCAUSER.

   z/OS **ALTMCA (solamente z/OS)**
   Como en CTX, pero no se utiliza ningún ID de usuario recibido de la red.

Autorización de transferencia de mensajes a colas de clústeres remotos

En z/OS , configure la autorización para transferir a una cola de clúster utilizando RACF. En Multiplatforms, autorice el acceso para conectarse a los gestores de colas y para colocar en las colas de dichos gestores de colas.

Acerca de esta tarea

El comportamiento predeterminado es realizar el control de acceso para SYSTEM.CLUSTER.TRANSMIT.QUEUE. Tenga en cuenta que este comportamiento se aplica, incluso si está utilizando varias colas de transmisión.

El comportamiento descrito en este tema solamente se aplica si ha configurado el atributo **ClusterQueueAccessControl** en el archivo qm.ini para que sea *RQMName*, tal como se describe en el tema Stanza de seguridad y si ha reiniciado el gestor de colas.

Procedimiento

- z/OS
  Para z/OS, emita los mandatos siguientes:

  ```
  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 los mandatos siguientes:

```
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 los mandatos siguientes:

```
GRTMQMAUT OBJ(' QMgrName ') OBJTYPE(*MQM) USER(GroupName) AUT(*CONNECT)
GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

El usuario sólo puede transferir mensajes a la cola de clúster especificada, y no a otras colas de clúster.

Los nombres de las variables tienen los significados siguientes:

QMgrName
Nombre del gestor de colas. En z/OS, este valor también puede ser el nombre de un grupo de compartición de colas.

GroupName
Nombre del grupo al que se va a otorgar el acceso.

QueueName
Nombre de la cola o perfil genérico para el que se van a cambiar autorizaciones.

Qué hacer a continuación

Si especifica una cola de respuesta cuando transfiere un mensaje a una cola de clúster, la aplicación de consumo debe tener autorización para enviar la respuesta. Establezca esta autorización siguiendo las instrucciones de “Otorgar autorización para transferir mensajes a una cola de clúster remota” en la página 403.

Conceptos relacionados
Stanza de seguridad en qm.ini

Impedir que gestores de colas se unan a un clúster

Si un gestor de colas falso se une a un clúster, es difícil impedir que reciba mensajes que usted no desea que reciba.

Procedimiento

Si desea asegurarse de que sólo determinados gestores de colas autorizados se unen a un clúster, puede elegir entre tres técnicas:

- Mediante el uso de registros de autenticación de canal, puede bloquear la conexión de canal de clúster basándose en: la dirección IP remota, el nombre del gestor de colas remoto o el Nombre distinguido TLS proporcionado por el sistema remoto.
- Escribir un programa de salida para impedir que los gestores de colas no autorizados graben en la cola SYSTEM.CLUSTER.COMMAND.QUEUE. No restrinja el acceso a SYSTEM.CLUSTER.COMMAND.QUEUE de manera que ningún gestor de colas pueda grabar en ella, o impedirá que cualquier gestor de colas que se una al clúster.
- Un programa de salida de seguridad en la definición de canal CLUSRCVR.

Salidas de seguridad en canales de clúster

Consideraciones adicionales al utilizar salidas de seguridad en canales de clúster.

Acerca de esta tarea

Cuando un canal de clúster emisor se inicia por primera vez, utiliza atributos definidos manualmente por un administrador del sistema. Cuando el canal se detiene y se reinicia, toma los atributos de la definición de canal de clúster emisor correspondiente. La definición de canal de clúster emisor original se sobrescribe con los nuevos atributos, incluido el atributo `SecurityExit`.

Procedimiento

1. Debe definir una salida de seguridad tanto en el extremo del clúster emisor como en el extremo del clúster receptor de un canal.

   La conexión inicial debe establecerse con un reconocimiento de salida de seguridad, aunque el nombre de salida de seguridad se envíe desde la definición de clúster receptor.
2. Valide el `PartnerName` en la estructura `MQCXP` de la salida de seguridad.

   La salida debe permitir que el canal se inicie únicamente si el gestor de colas asociado está autorizado.
3. Diseñe la salida de seguridad de la definición de clúster receptor para que se inicie con el receptor.
4. Si la diseña como iniciada con el emisor, un gestor de colas no autorizado sin una salida de seguridad puede unirse al clúster porque no se realiza ninguna comprobación de seguridad.

   Hasta que el canal no se haya detenido y reiniciado, no se podrá enviar el nombre `SCYEXIT` desde la definición de clúster receptor ni se podrán realizar comprobaciones de seguridad completas.
5. Para ver la definición de canal de clúster emisor que se está utilizando en este momento, utilice el mandato:

   ```
   DISPLAY CLUSQMGR( queue manager ) ALL
   ```

   El mandato muestra los atributos que se han enviado desde la definición de clúster receptor.
6. Para ver la definición original, utilice el mandato:

   ```
   DISPLAY CHANNEL( channel name ) ALL
   ```
7. Es posible que tenga que definir una salida de definición automática de canal, `CHADEXIT`, en el gestor de colas del clúster emisor, si los gestores de colas se encuentran en plataformas diferentes.

   Utilice la salida de definición automática de canal para establecer el atributo `SecurityExit` en un formato adecuado para la plataforma de destino.
8. Despliegue y configure la salida de seguridad.

   z/OS z/OS

   El módulo de carga de salida de seguridad debe estar en el conjunto de datos especificado en la sentencia `CSQXLIB DD` del procedimiento de espacio de direcciones del iniciador de canal.

   ALW AIX, Linux, and Windows sistemas

   - La biblioteca de enlace dinámico de salida de seguridad debe estar en la vía de acceso especificada en el atributo `SCYEXIT` de la definición de canal.
   - La biblioteca de enlace dinámico de salida de definición automática de canal debe estar en la vía de acceso especificada en el atributo `CHADEXIT` de la definición de gestor de colas.

Forzar que los gestores de colas no deseados abandonen un clúster

Puede forzar que un gestor de colas no deseado abandone un clúster emitiendo el mandato RESET CLUSTER en un gestor de colas de repositorio completo.

Acerca de esta tarea

Puede forzar a que un gestor de colas no deseado deje un clúster. Por ejemplo, si se suprime un gestor de colas pero sus canales de clúster receptor siguen estando definidos en el clúster, es posible que desee una reorganización.

Sólo los gestores de colas de repositorio completo tienen autorización para expulsar a un gestor de colas de un clúster.

Nota: Aunque la utilización del mandato RESET CLUSTER fuerza la eliminación de un gestor de colas de un clúster, si utiliza RESET CLUSTER por sí solo no impide que el gestor de colas se reincorpore al clúster más adelante. Para asegurarse de que el gestor de colas no se vuelva a unir al clúster, siga los pasos detallados en "Impedir que gestores de colas se unan a un clúster" en la página 488.

Siga este procedimiento para expulsar al gestor de colas OSLO del clúster NORWAY:

Procedimiento

1. En un gestor de colas de depósito completo, emita el mandato:

```
RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
```

2. O utilice el MQID en lugar de QMNAME en el mandato:

```
RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
```

   Nota: QMID es una serie, por lo que el valor de qmid debe estar entre comillas simples, por ejemplo, QMID('FR01_2019-07-15_14.42.42').

Resultados

El gestor de colas que se elimina forzadamente no cambia; sus definiciones de clúster locales muestran que está en el clúster. Las definiciones en todos los demás gestores de colas no muestran que está en el clúster.

Cómo impedir que los gestores de colas reciban mensajes

Puede impedir que un gestor de colas reciba mensajes si no está autorizado para recibirlos utilizando programas de salida.

Acerca de esta tarea

Es difícil impedir a un gestor de colas de un clúster que defina una cola. Existe el peligro de que un gestor de colas falso pueda unirse a un clúster y defina su propia instancia de una de las colas en el clúster. Ahora puede recibir mensajes que no está autorizado a recibir. Para impedir que un gestor de colas reciba mensajes, utilice una de las opciones siguientes indicadas en el procedimiento.

Procedimiento

- Un programa de salida de canal en cada canal de clúster emisor. El programa de salida utiliza el nombre de conexión para determinar la adecuación del gestor de colas de destino al que se deban enviar los mensajes.
- Un programa de salida de carga de trabajo del clúster, que utiliza los registros de destino para determinar la adecuación de la cola de destino y el gestor de colas al que se deban enviar los mensajes.

SSL/TLS y clústeres

Al configurar TLS para clústeres, tenga en cuenta que se propaga una definición de canal CLUSRCVR a otros gestores de colas como un canal CLUSSDR definido automáticamente. Si un canal CLUSRCVR utiliza TLS, debe configurar TLS en todos los gestores de colas que se comuniquen utilizando el canal.

Para obtener más información sobre TLS, consulte "Protocolos de seguridad TLS en IBM MQ" en la página 25. Los consejos que se ofrecen en dicho tema generalmente son aplicables a los canales del clúster, pero tal vez desee considerar lo siguiente:

En un clúster de IBM MQ se propaga frecuentemente una definición de canal CLUSRCVR específica a muchos otros gestores de colas, donde se transforma en un CLUSSDR definido automáticamente. Posteriormente, el CLUSSDR definido automáticamente se utiliza para iniciar un canal para el CLUSRCVR. Si el CLUSRCVR está configurado para la conectividad TLS, se aplican las siguientes consideraciones:

- Todos los gestores de colas que deseen comunicarse con este CLUSRCVR debe tener acceso al soporte de TLS. Esta provisión de TLS debe dar soporte a la CipherSpec para el canal.
- Los diferentes gestores de colas a los que se han propagado los canales de clúster emisor definidos automáticamente tendrán cada uno un nombre distinguido diferente asociado. Si se va a utilizar la comprobación de nombres distinguidos de iguales en el CLUSRCVR, éste debe configurare de manera que todos los nombres distinguidos que puedan recibirse se comparen correctamente.

  Por ejemplo, supongamos que todos los gestores de colas que alojarán canales de clúster emisor que conectarán a un CLUSRCVR determinado, tienen certificados asociados. Supongamos también que los nombres distinguidos en todos estos certificados definen el país como UK, la organización como IBM, la unidad organizativa como IBM MQ Development, y que todos tienen nombres comunes en el formato DEVT.QMnnn, donde nnn es un valor numérico.

  En este caso, un valor de SSLPEER de `C=UK, O=IBM, OU=IBM MQ Development, CN=DEVT.QM*` en el CLUSRCVR permitirá que todos los canales de emisor de clúster se conecten correctamente, pero impedirá que se conecten canales de emisor de clúster no deseados.
- Si se utilizan series CipherSpec personalizadas, tenga en cuenta que los formatos de serie personalizados no están permitidos en todas las plataformas. Un ejemplo de esto es que la CipherSpec serie RC4_SHA_US tiene un valor de 05 en IBM i pero no es una especificación válida en sistemas AIX, Linux, and Windows . Por lo tanto, si se utilizan parámetros SSLCIPH personalizados en un CLUSRCVR, todos los canales de clúster emisor definidos automáticamente resultantes deben residir en plataformas en las que el soporte TLS subyacente implemente esta CipherSpec y en las que se pueda especificar con el valor personalizado. Si no puede seleccionar un valor para el parámetro SSLCIPH que se pueda entender en todo el clúster, necesitará una salida de definición automática de canal para transformarla en algo que puedan interpretar las plataformas que se utilizan. Utilice las series CipherSpec de texto cuando sea posible (por ejemplo TLS_RSA_WITH_AES_128_CBC_SHA).

Un parámetro SSLCRLNL se aplica a un gestor de colas individual y no se propaga a otros gestores de colas de un clúster.

Actualización de gestores de colas y canales en clúster a SSL/TLS

Actualice los canales de clúster de uno en uno, cambiando todos los canales CLUSRCVR antes que los canales CLUSSDR.

Antes de empezar

Tenga en cuenta las consideraciones siguientes, ya que estas podrían afectar a la elección de CipherSpec para un clúster:

- Algunas CipherSpecs no están disponibles en todas las plataformas. Procure elegir una CipherSpec que esté soportada por todos los gestores de colas en el clúster.
- Algunas CipherSpecs podrían ser nuevas en el release de IBM MQ actual y no se soportan en releases anteriores. Un clúster que contiene gestores de colas que se ejecutan en releases MQ diferentes sólo podrá utilizar las CipherSpecs soportadas por cada release.

Para utilizar una nueva CipherSpec dentro de un clúster, primero debe migrar todos los gestores de colas de clúster al release actual.

- Algunas CipherSpecs requieren el uso de un tipo específico de certificado digital, especialmente aquellas que utilizan cifrado Elliptic Curve.

Atención: No es posible utilizar una combinación de certificados firmados por Elliptic Curve y los certificados firmados por RSA en los gestores de colas que desea unir como parte de un clúster.

Los gestores de colas de un clúster deben utilizar todos los certificados firmados por RSA, o bien utilizar todos los certificados firmados por EC, no una combinación de ambos.

Consulte "Certificados digitales y compatibilidad de CipherSpec en IBM MQ" en la página 49 para obtener más información.

Actualice todos los gestores de colas del clúster a IBM MQ V8 o superior, si todavía no están en estos niveles. Distribuya los certificados y las claves para que TLS funcione desde cada uno de ellos.

Para poder actualizar o utilizar cualquiera de los alias CipherSpecs (ANY_TLS13, ANY_TLS13_OR_HIGHER, ANY_TLS12, ANY_TLS12_OR_HIGHER, etc.), debe actualizar los gestores de colas:

- Multi Actualice todos los gestores de colas de IBM MQ for Multiplatforms del clúster a IBM MQ 9.1.4 o posterior.
- z/OS Actualice todos los gestores de colas de IBM MQ for z/OS del clúster a IBM MQ for z/OS 9.2.0 o posterior.

debe

Acerca de esta tarea

Cambie los canales CLUSRCVR antes que los canales CLUSSDR.

Procedimiento

1. Conmute los canales CLUSRCVR a TLS en cualquier orden que desee, cambiando los canales CLUSRCVR de uno en uno, y permita que los cambios circulen por el clúster antes de cambiar el siguiente.

   Importante: Asegúrese de no cambiar la ruta inversa hasta que los cambios para el canal actual se hayan distribuido por el clúster.
2. Opcional: Cambie todos los canales CLUSSDR manuales a TLS.

   Esto no tiene ningún efecto en el funcionamiento del clúster, a menos que se utilice el mandato REFRESH CLUSTER con la opción REPOS(YES).

   Nota: Para clústeres de gran tamaño, el uso del mandato **REFRESH CLUSTER** puede interrumpir el clúster mientras está en curso, y de nuevo a intervalos de 27 días a partir de entonces cuando los objetos de clúster envían automáticamente actualizaciones de estado a todos los gestores de colas interesados. Consulte La renovación en un clúster grande puede afectar el rendimiento y la disponibilidad del clúster.
3. Utilice el mandato DISPLAY CLUSQMGR para asegurarse de que la nueva configuración de seguridad se ha propagado en todo el clúster.
4. Reinicie los canales para que utilicen TLS y ejecute REFRESH SECURITY (SSL).

Conceptos relacionados

"Habilitación de CipherSpecs" en la página 428
Habilite una CipherSpec utilizando el parámetro **SSLCIPH** en el mandato **DEFINE CHANNEL** o **ALTER CHANNEL** MQSC.

"Certificados digitales y compatibilidad de CipherSpec en IBM MQ" en la página 49

En este tema se proporciona información sobre cómo elegir las CipherSpecs y los certificados digitales adecuados para su política de seguridad, describiendo la relación entre las CipherSpecs y los certificados digitales en IBM MQ.

Información relacionada

Agrupación en clúster: utilización de las recomendaciones de REFRESH CLUSTER

Inhabilitación de SSL/TLS en gestores de colas y canales en clúster

Para desactivar TLS, establezca el parámetro SSLCIPH en `' '`. Inhabilite TLS en los canales del clúster de individual, cambiando todos los canales de clúster receptores antes que los canales de clúster emisores.

Acerca de esta tarea

Cambie un canal de clúster emisor cada vez y permita que los cambios fluyan por el clúster antes de cambiar el siguiente.

Importante: Asegúrese de no cambiar la ruta inversa hasta que los cambios para el canal actual se hayan distribuido por el clúster.

Procedimiento

1. Defina el valor del parámetro SSLCIPH en `' '`, una serie vacía entre comillas sencillas IBM i, o *NONE en IBM i .

   Puede desactivar TLS en los canales de clúster receptores en el orden que desee.

   Tenga en cuenta que los cambios fluyen en dirección opuesta por canales en los que se deja TLS activo.
2. Compruebe que el nuevo valor se refleja en todos los demás gestores de colas utilizando el mandato **DISPLAY CLUSQMGR(*)** ALL.
3. Desactive TLS en todos los canales de clúster emisores manuales.

   Esto no tiene ningún efecto en el funcionamiento del clúster, a menos que se utilice el mandato **REFRESH CLUSTER** con la opción REPOS(YES).

   Para los clústeres de gran tamaño, utilice el mandato **REFRESH CLUSTER** puede generar problemas a intervalos regulares posteriormente, cuando los objetos de clúster envían automáticamente actualizaciones de estado a todos los gestores de colas interesados. Consulte La actualización en un clúster de gran tamaño puede afectar al rendimiento y la disponibilidad del clúster para obtener más información.
4. Detenga y reinicie los canales de clúster emisores.

Seguridad de publicación/suscripción

Los componentes e interacciones que están implicados en la publicación/suscripción se describen como una introducción a las explicaciones más detalladas y los ejemplos que siguen.

Hay una serie de componentes implicados en la publicación y suscripción a un tema. Algunas de las relaciones de seguridad entre ellos se ilustran en la Figura 22 en la página 494 y se describen en el siguiente ejemplo.

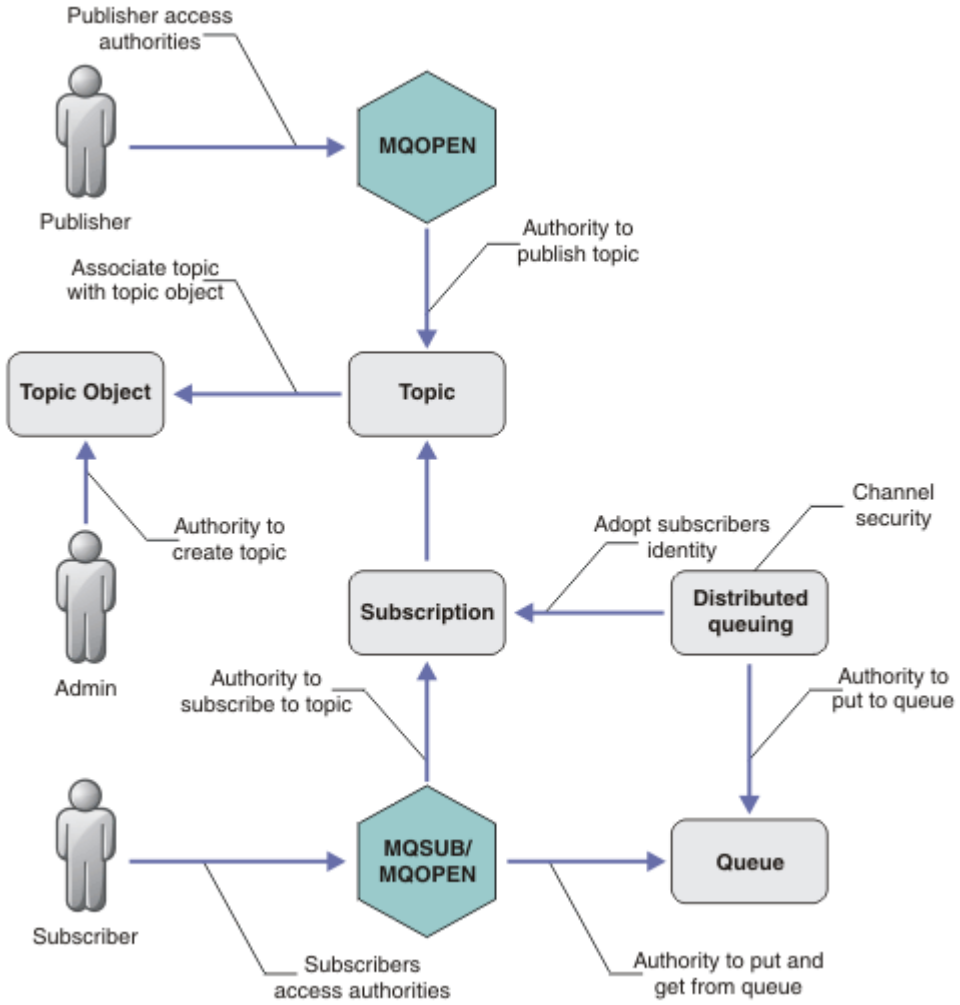


Figura 22. Relaciones de seguridad de publicación/suscripción entre diversos componentes

Temas

Los temas se identifican mediante series de tema, y normalmente se organizan en árboles; consulte Árboles de temas. Debe asociar un tema con un objeto de tema para controlar el acceso al tema. En la sección "Modelo de seguridad de temas" en la página 496 se describe cómo proteger los temas mediante objetos de tema.

Objetos de temas administrativos

Puede controlar quién tiene acceso a un tema, y con qué finalidad, mediante el mandato **setmqaut** con una lista de objetos de temas administrativos. Consulte los ejemplos en "Otorgar acceso a un usuario para suscribirse a un tema" en la página 501 y en "Otorgar acceso a un usuario para publicar en un tema" en la página 508.

z/OS Para controlar el acceso a los objetos de tema en z/OS, consulte Perfiles para la seguridad de tema.

Suscripciones

Suscríbase a uno o varios temas creando una suscripción mediante una serie de tema, que puede incluir comodines, para que coincida con la serie de tema de las publicaciones. Para obtener más detalles, consulte:

Suscripción mediante un objeto de tema

"Suscripción utilizando el nombre de objeto de tema" en la página 497

Suscripción mediante un tema
"Suscripción utilizando una serie del tema donde el nodo de tema no existe" en la página 498

Suscripción mediante un tema con comodines
"Suscripción utilizando una serie de tema que contiene caracteres comodín" en la página 498

Una suscripción contiene información sobre la identidad del suscriptor y la identidad de la cola de destino en la que se van a colocar las publicaciones. También contiene información sobre cómo debe colocarse la publicación en la cola de destino.

Del mismo modo que puede definir qué suscriptores tienen autorización para suscribirse a determinados temas, puede restringir las suscripciones para que sean utilizadas por un suscriptor individual. También puede controlar qué información sobre el suscriptor utiliza el gestor de colas cuando las publicaciones se colocan en la cola de destino. Consulte "Seguridad de suscripción" en la página 514.

Colas

La cola de destino es una cola importante que debe protegerse. Es local para el suscriptor, y las publicaciones que coinciden con la suscripción se colocan en ella. Debe tener en cuenta el acceso a la cola de destino desde dos perspectivas:

1. Transferencia de una publicación a la cola de destino.
2. Obtención de la publicación de la cola de destino.

El gestor de colas transfiere una publicación a la cola de destino utilizando una identidad proporcionada por el suscriptor. El suscriptor, o un programa al que ha sido delegado la tarea de obtener publicaciones, toma mensajes de la cola. Consulte "Autorización para colas de destino" en la página 499.

No hay alias de objeto de tema, pero puede utilizar una cola de alias como alias para un objeto de tema. Si lo hace, y se comprueba la autorización para utilizar el tema de publicación o suscripción, el gestor de colas comprueba la autorización para utilizar la cola.

"Seguridad de publicación/suscripción entre gestores de colas" en la página 515

Su permiso para publicar o suscribirse a un tema se comprueba en el gestor de colas local utilizando identidades y autorizaciones locales. La autorización no depende de si el tema se define o no, ni de dónde está definido. Por consiguiente, debe realizar la autorización de temas en cada gestor de colas de un clúster cuando se utilizan temas en clúster.

Nota: El modelo de seguridad para los temas difiere del modelo de seguridad para las colas. Puede conseguir el mismo resultado para las colas mediante la definición, a nivel local, de un alias de cola para cada cola en clúster.

Los gestores de colas intercambian suscripciones en un clúster. En la mayoría de configuraciones de clúster de IBM MQ, los canales se configuran con PUTAUT=DEF para colocar mensajes en las colas de destino usando la autorización del proceso del canal. Se puede modificar la configuración del canal para utilizar PUTAUT=CTX a fin de exigir que el usuario suscriptor tenga autorización para propagar una suscripción a otro gestor de colas en un clúster.

En "Seguridad de publicación/suscripción entre gestores de colas" en la página 515 se describe cómo cambiar las definiciones de canal para controlar quién tiene permiso para propagar suscripciones en otros servidores del clúster.

Autorización

Puede aplicar autorización a objetos de tema, como ocurre con las colas y otros objetos. Hay tres operaciones de autorización, `pub`, `sub` y `resume` que pueden aplicarse sólo a temas. Los detalles se describen en Especificación de autorizaciones para tipos de objeto diferentes.

Llamadas de función

En programas de publicación y suscripción, como en programas de transmisión a colas, las comprobaciones de autorización se realizan cuando se abren, crean, cambian o eliminan objetos. No se realizan comprobaciones cuando se llevan a cabo llamadas MQPUT o MQGET de MQI para transferir y obtener publicaciones.

Para publicar un tema, realice una llamada MQOPEN en el tema, que realiza las comprobaciones de autorización. Publique mensajes en el descriptor de tema mediante el mandato MQPUT, que no realiza comprobaciones de autorización.

Para suscribirse a un tema, generalmente debe ejecutar un mandato MQSUB para crear o reanudar una suscripción, y también abrir la cola de destino para que pueda recibir publicaciones. De forma alternativa, ejecute un mandato MQOPEN por separado para abrir la cola de destino y, a continuación, ejecute un mandato MQSUB para crear o reanudar la suscripción.

Independientemente de las llamadas que utilice, el gestor de colas comprueba que puede suscribirse al tema y obtener las publicaciones resultantes de la cola de destino. Si la cola de destino no está gestionada, también se realizan comprobaciones de la autorización para ver si el gestor de colas puede transferir publicaciones a la cola de destino. Utiliza la identidad que adoptó a partir en una suscripción coincidente. Se supone que el gestor de colas siempre es capaz de colocar las publicaciones en las colas de destino gestionado.

Roles

Los usuarios están involucrados en cuatro roles al ejecutar aplicaciones de publicación/suscripción:

1. Publicador
2. Suscriptor
3. Administrador de temas
4. Administrador de IBM MQ, miembro del grupo mqm

Defina grupos con las autorizaciones apropiadas que correspondan a los roles de publicación, suscripción y administración de temas. A continuación, puede asignar principales a estos grupos autorizándoles a realizar tareas específicas de publicación y suscripción.

Además, debe ampliar las autorizaciones de operaciones administrativas para el administrador de colas y canales responsable de mover publicaciones y suscripciones.

Modelo de seguridad de temas

Los objetos de tema definidos son los únicos que pueden tener atributos de seguridad asociados. Para obtener una descripción de los objetos de tema, consulte Objetos de tema administrativo. Los atributos de seguridad especifican si un ID de usuario determinado, o un grupo de seguridad, pueden realizar una operación de suscripción o publicación en cada objeto de tema.

Los atributos de seguridad están asociados con el nodo de administración adecuado en el árbol de temas. Cuando se efectúa una comprobación de autorización para un ID de usuario determinado durante una operación de suscripción o publicación, la autorización otorgada se basa en los atributos de seguridad del nodo del árbol de temas asociado.

Los atributos de seguridad son una lista de control de acceso que indica qué autorización tiene un ID de usuario o grupo de seguridad determinado del sistema operativo sobre el objeto de tema.

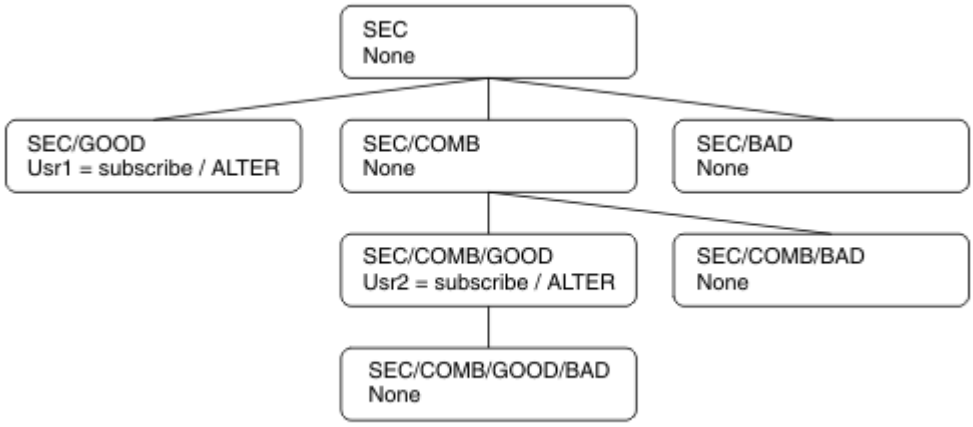
Considere el ejemplo siguiente donde los objetos de tema se han definido con atributos de seguridad o autorizaciones:

Tabla 87. Ejemplo de autorizaciones de objetos de tema

Nombre de tema	Serie de tema	Autoridades-Multiplataformas	Autorizaciones de z/OS
SECROOT	SEC	Ninguna	Ninguna
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD
SECBAD	SEC/BAD	Ninguna	Ninguna HLQ.SUBSCRIBE.SECBAD

Tabla 87. Ejemplo de autorizaciones de objetos de tema (continuación)			
Nombre de tema	**Serie de tema**	**Autoridades-Multiplataformas**	**Autorizaciones de z/OS**
SECCOMB	SEC/COMB	Ninguna	Ninguna HLQ.SUBSCRIBE.SECCOMB
SECCOMBB	SEC/COMB/GOOD/BAD	Ninguna	Ninguna HLQ.SUBSCRIBE.SECCOMBB
SECCOMBG	SEC/COMB/GOOD	usr2+subscribe	ALTER HLQ.SUBSCRIBE.SECCOMBG
SECCOMBN	SEC/COMB/BAD	Ninguna	Ninguna HLQ.SUBSCRIBE.SECCOMBN

El árbol de temas con los atributos de seguridad asociados en cada nodo puede representar del siguiente modo:



Los ejemplos enumerados otorgan las autorizaciones siguientes:

- En el nodo raíz del árbol de /SEC, ningún usuario tiene autorización en dicho nodo.
- A usr1 se le ha otorgado autorización de suscripción para el objeto /SEC/GOOD
- A usr2 se le ha otorgado autorización de suscripción para el objeto /SEC/COMB/GOOD

Suscripción utilizando el nombre de objeto de tema

Al suscribirse a un objeto de tema especificando el nombre MQCHAR48, se localiza el nodo correspondiente del árbol de temas. Si los atributos seguridad asociados con el nodo indican que el usuario tiene autorización para suscribirse, se otorga acceso.

Si no se otorga acceso al usuario, el nodo padre del árbol determina si el usuario tiene autorización para suscribirse en el nivel de nodo padre. Si es así, se otorga acceso. En caso contrario, se considera el padre de dicho nodo. La recurrencia continúa hasta que se encuentra un nodo que otorga autorización de suscripción al usuario. La recurrencia se detiene cuando se considera el nodo raíz sin haber sido otorgado autorización. En este último caso, se deniega el acceso.

En pocas palabras, si cualquier nodo en la vía otorga al usuario o aplicación autorización para suscribirse, el suscriptor está autorizado para suscribirse a dicho nodo, o a cualquier nodo por debajo de dicho nodo en el árbol de temas.

El nodo raíz en el ejemplo es SEC.

Se otorga autorización de suscripción al usuario si la lista de control de acceso indica que el propio ID de usuario tiene autorización, o que un grupo de seguridad del sistema operativo del que el ID de usuario es miembro tiene autorización.

Así, por ejemplo:

- Si `usr1` intenta suscribirse mediante una serie de tema de `SEC/GOOD`, la suscripción se permitirá porque el ID de usuario tiene acceso al nodo asociado con dicho tema. Sin embargo, si `usr1` ha intentado suscribirse utilizando la serie de tema `SEC/COMB/GOOD`, la suscripción no se permitirá porque el ID de usuario no tiene acceso a la nodo asociado a él.
- Si `usr2` intenta suscribirse mediante una serie de tema de `SEC/COMB/GOOD`, la suscripción se permitirá porque el ID de usuario tiene acceso al nodo asociado con el tema. Sin embargo, si `usr2` ha intentado suscribirse a `SEC/GOOD`, la suscripción no se permitirá porque el ID de usuario no tiene acceso al nodo asociado a él.
- Si `usr2` intenta suscribirse utilizando una serie de tema de `SEC/COMB/GOOD/BAD`, la suscripción se permitirá porque el ID de usuario tiene acceso al nodo padre `SEC/COMB/GOOD`.
- Si `usr1` o `usr2` intentan suscribirse utilizando una serie de tema de `/SEC/COMB/BAD`, no se permitirá porque no tienen acceso al nodo de tema asociado o a los nodos padre de dicho tema.

Una operación de suscripción que especifique el nombre de un objeto de tema que no existe dará lugar a un error MQRC_UNKNOWN_OBJECT_NAME.

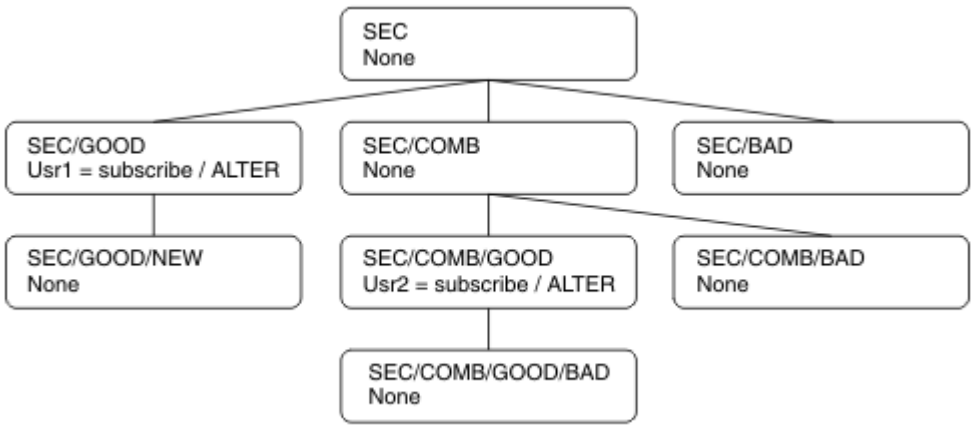
Suscripción utilizando una serie del tema donde el nodo de tema existe

El comportamiento es el mismo que cuando se especifica el tema por el nombre del objeto MQCHAR48.

Suscripción utilizando una serie del tema donde el nodo de tema no existe

Considere el caso de una aplicación de suscripción que especifica una serie de tema que representa un nodo de tema que no existe actualmente en el árbol de temas. La comprobación de autorización se realiza como se describe en el apartado anterior. La selección empieza con el nodo padre representado por la serie de tema. Si se otorga la autorización, se crea un nodo nuevo que representa la serie de tema en el árbol de temas.

Por ejemplo, `usr1` intenta suscribirse a un tema `SEC/GOOD/NEW`. La autorización se otorga porque `usr1` tiene acceso al nodo padre `SEC/GOOD`. Se crea un nodo de tema nuevo en el árbol como se muestra en el siguiente diagrama. El nodo de tema nuevo no es un objeto de tema y no tiene ningún atributos de seguridad asociado directamente; los atributos los hereda de su padre.



Suscripción utilizando una serie de tema que contiene caracteres comodín

Considere el caso de una suscripción mediante una serie de tema que contiene un carácter comodín. La comprobación de autorización se efectúa en el nodo del árbol de temas que coincide con la parte completa de la serie de tema.

Por lo tanto, si una aplicación se suscribe a SEC/COMB/GOOD/*, se lleva a cabo una comprobación de autorización como se describe en las dos secciones anteriores en el nodo SEC/COMB/GOOD del árbol de temas.

Del mismo modo, si una aplicación debe suscribirse a SEC/COMB/*/GOOD, se lleva a cabo una comprobación de autorización en el nodo SEC/COMB.

Autorización para colas de destino

Al suscribirse a un tema, uno de los parámetros es el manejador hobj de una cola que se ha abierto para salida para recibir las publicaciones.

Si no se especifica hobj, pero está en blanco, se crea una cola gestionada si se cumplen las condiciones siguientes:

- Se ha especificado la opción MQSO_MANAGED.
- La suscripción no existe.
- Se ha especificado creación.

Si deja hobj en blanco, y modifica o reanuda una suscripción existente, la cola de destino indicada anteriormente debe ser gestionada o no gestionada.

La aplicación o el usuario que realiza la solicitud de MQSUB debe tener la autorización para transferir mensajes a la cola de destino especificada; en efecto, debe tener la autorización para transferir mensajes publicados a esa cola. La comprobación de autorización sigue las reglas existentes para la comprobación de la seguridad de las colas.

La comprobación de seguridad incluye el ID de usuario alternativo y las comprobaciones de seguridad de contexto si es necesario. Para poder establecer cualquiera de los campos de contexto de identidad, debe especificar la opción MQSO_SET_IDENTITY_CONTEXT, así como la opción MQSO_CREATE o MQSO_ALTER. No se puede establecer ninguno de los campos de contexto de identidad en una solicitud MQSO_RESUME.

Si el destino es una cola gestionada, no se realiza ninguna comprobación de seguridad en el destino gestionado. Si se le permite suscribirse a un tema, se supone que puede utilizar destinos gestionados.

Publicación utilizando el nombre de tema o una serie de tema donde el nodo de tema existe

El modelo de seguridad de la publicación es el mismo que el de la suscripción, excepto los caracteres comodín. Las publicaciones no contienen comodines; por lo tanto no hay ningún caso de una serie de tema que contenga caracteres comodín para tener en cuenta.

Las autorizaciones para publicar y suscribir son diferentes. Un usuario o grupo puede tener la autorización para llevar a cabo una de estas operaciones sin que necesariamente pueda realizar la otra.

Cuando se publica en un objeto de tema especificando el nombre MQCHAR48 o la serie del tema, se localiza el nodo correspondiente del árbol de temas. Si los atributos de seguridad asociados con el nodo de tema indican que el usuario tiene autorización para publicar, se otorga acceso.

Si no se ha otorgado el acceso, el nodo padre en el árbol determina si el usuario tiene autorización para publicar en dicho nivel. Si es así, se otorga acceso. Si no, la recurrencia continúa hasta que se encuentra un nodo que otorgue autorización de publicación para el usuario. La recurrencia se detiene cuando se considera el nodo raíz sin haber sido otorgado autorización. En este último caso, se deniega el acceso.

En pocas palabras, si algún nodo de la vía otorga a dicho usuario o aplicación autorización para publicar, el publicador puede publicar en dicho nodo o en cualquier lugar bajo dicho nodo en el árbol de temas.

Publicación utilizando el nombre de tema o una serie de tema donde el nodo de tema no existe

Como ocurre con la operación de suscripción, cuando una aplicación publica especificando una serie de tema que representa un nodo de tema que no existe actualmente en el árbol de temas, la comprobación

de autorización se realiza empezando por el padre del nodo representado por la serie de tema. Si se otorga la autorización, se crea un nodo nuevo que representa la serie de tema en el árbol de temas.

Publicación utilizando una cola de alias que se resuelve en un objeto de tema

Si publica utilizando una cola de alias que se resuelve en un objeto de tema, la comprobación de seguridad se produce tanto en la cola de alias como en el tema subyacente en el que se resuelve.

La comprobación de seguridad en la cola de alias verifica que el usuario tiene autorización para transferir mensajes a esa cola de alias y la comprobación de seguridad sobre el tema verifica que el usuario puede publicar en dicho tema. Cuando una cola alias se resuelve en otra cola, las comprobaciones no se realizan en la cola subyacente. La comprobación de autorización se realiza de forma distinta para temas y colas.

Cierre de una suscripción

Existe una comprobación de seguridad adicional si se cierra una suscripción utilizando la opción MQCO_REMOVE_SUB y si no se ha creado la suscripción bajo este manejador.

Se realiza una comprobación de seguridad para garantizar que tiene la autorización correcta para hacerlo, porque la acción da como resultado la eliminación de la suscripción. Si los atributos de seguridad asociados con el nodo de tema indican que el usuario tiene autorización, se otorga acceso. Si no es así, se considera el nodo padre del árbol para determinar si el usuario tiene autorización para cerrar la suscripción. La recurrencia continúa hasta que se otorga autorización o bien hasta que se alcanza el nodo raíz.

Definición, modificación y supresión de una suscripción

No se lleva a cabo ninguna comprobación de seguridad de la suscripción cuando se crea una suscripción administrativamente, en lugar de utilizar una solicitud de API MQSUB. El administrador ya ha recibido esta autorización a través del mandato.

Las comprobaciones de seguridad se realizan para garantizar que las publicaciones se pueden transferir a la cola de destino asociada con la suscripción. Las comprobaciones se realizan del mismo modo que para una solicitud MQSUB.

El ID de usuario que se utiliza para estas comprobaciones de seguridad depende del mandato que se emite. Si se especifica el parámetro **SUBUSER**, ello afecta al modo en que se lleva a cabo la comprobación, como se muestra en Tabla 88 en la página 500:

Tabla 88. ID de usuario utilizados para las comprobaciones de seguridad para mandatos

Mandato	SUBUSER especificado y en blanco	SUBUSER especificado y completo	SUBUSER no especificado
	Utilizar el ID de administrador		Utilizar el ID de usuario de la suscripción LIKE
	Utilizar el ID de administrador		Utilizar el ID.DEFAULT.SU de usuarioB - si está en de lablanco, suscripciónutilizar el ID SYSTEMde administrador

Tabla 88. ID de usuario utilizados para las comprobaciones de seguridad para mandatos (continuación)			
Mandato	**SUBUSER especificado y en blanco**	**SUBUSER especificado y completo**	**SUBUSER no especificado**
	Utilizar el ID de administrador		Utilizar el ID de usuario de la suscripción existente

La única comprobación de seguridad que se lleva a cabo al suprimir las suscripciones con el mandato DELETE SUB es la comprobación de seguridad de mandatos.

Ejemplo de configuración de seguridad de publicación/suscripción

En esta sección se describe un escenario que tiene configurado el control de accesos a los temas de forma que permite aplicar el control de seguridad según sea necesario.

Otorgar acceso a un usuario para suscribirse a un tema

Este tema es el primero de una lista de tareas que indica cómo otorgar acceso a los temas por más de un usuario.

Acerca de esta tarea

En esta tarea se presupone que no existen objetos de temas administrativos, ni se han definido los perfiles para la suscripción o publicación. Las aplicaciones crean nuevas suscripciones, en lugar de reanudar las existentes, y lo hacen utilizando sólo la serie de tema.

Una aplicación puede realizar una suscripción proporcionando un objeto de tema, o una serie de tema, o una combinación de ambos. Sea cual sea lo que seleccione la aplicación, el efecto es crear una suscripción en un punto determinado del árbol de temas. Si este punto del árbol de temas está representado por un objeto de tema administrativo, se comprueba un perfil de seguridad según el nombre de este objeto de tema.

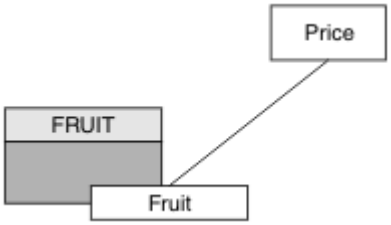


Figura 23. Ejemplo de acceso a objeto de tema

Tabla 89. Ejemplo de acceso a objeto de tema		
Tema	**Acceso de suscripción necesario**	**Objeto de tema**
Price	Ningún usuario	Ninguna
Price/Fruit	USER1	FRUIT

Defina un nuevo objeto de tema de la manera siguiente:

Procedimiento

1. Emita el mandato MQSC `DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit')`.
2. Otorgue el acceso de la manera siguiente:

   - z/OS **z/OS** :

     Otorgue acceso a USER1 para suscribirse al tema `"Price/Fruit"` otorgando acceso de usuario al perfil `hlq.SUBSCRIBE.FRUIT`. Para ello, utilice los siguientes mandatos RACF:

     ```
     RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
     PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
     ```

   - Multi **Multiplataformas**:

     Otorgue acceso a USER1 para suscribirse al tema `"Price/Fruit"` otorgando acceso de usuario al objeto FRUIT. Hágalo mediante el mandato de autorización para la plataforma:

     ALW **AIX, Linux, and Windows sistemas**

     ```
     setmqaut -t topic -n FRUIT -p USER1 +sub
     ```

     IBM i **IBM i**

     ```
     GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
     ```

Resultados

Cuando USER1 intenta suscribirse al tema `"Price/Fruit"`, el resultado es satisfactorio.

Cuando USER2 intenta suscribirse al tema `"Price/Fruit"` el resultado es anómalo con un mensaje `MQRC_NOT_AUTHORIZED`, junto con:

- z/OS En z/OS, se ven los mensajes siguientes en la consola que muestran la vía de acceso de seguridad completa a través del árbol de temas que se ha intentado:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW En AIX, Linux, and Windows, el siguiente suceso de autorización:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit"
  ```

- IBM i En IBMi, el siguiente suceso de autorización:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit"
  ```

Tenga en cuenta que es una ilustración de lo que verá, no de todos los campos.

Otorgar acceso a un usuario para suscribirse a un tema más profundamente en el árbol

Este tema es el segundo de una lista de tareas que indica cómo otorgar acceso a los temas por más de un usuario.

Antes de empezar

Este tema utiliza la configuración descrita en "Otorgar acceso a un usuario para suscribirse a un tema" en la página 501.

Acerca de esta tarea

Si el punto del árbol de temas en que la aplicación realiza la suscripción no está representada por un objeto de tema administrativo, suba en el árbol hasta localizar el objeto de tema administrativo padre más cercano. El perfil de seguridad se comprueba, basándose en el nombre de dicho objeto de tema.

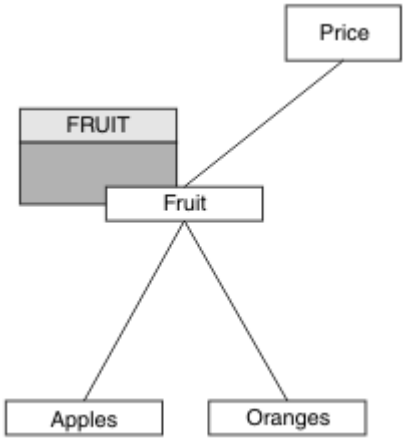


Figura 24. Ejemplo de otorgar acceso a un tema dentro de un árbol de temas

Tabla 90. Requisitos de acceso para los temas de ejemplo y los objetos de tema

Tema	Acceso de suscripción necesario	Objeto de tema
Price	Ningún usuario	Ninguna
Price/Fruit	USER1	FRUIT
Price/Fruit/Apples	USER1	
Price/Fruit/Oranges	USER1	

En "Otorgar acceso a un usuario para suscribirse a un tema" en la página 501, a USER1 se le ha otorgado acceso para suscribirse al tema `"Price/Fruit"` otorgándole acceso al perfil `hlq.SUBSCRIBE.FRUIT` en z/OS y acceso de suscripción de al perfil FRUIT en Multiplatforms. Este perfil único también otorga acceso a USER1 para suscribirse a `"Price/Fruit/Apples"`, `"Price/Fruit/Oranges"` y `"Price/Fruit/#"`.

Cuando USER1 intenta suscribirse al tema `"Price/Fruit/Apples"`, el resultado es satisfactorio.

Cuando USER2 intenta suscribirse al tema `"Price/Fruit/Apples"` el resultado es anómalo con un mensaje `MQRC_NOT_AUTHORIZED`, junto con:

- z/OS En z/OS, se ven los mensajes siguientes en la consola que muestran la vía de acceso de seguridad completa a través del árbol de temas que se ha intentado:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- Multi En Multiplatforms, el siguiente suceso de autorización:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

Tenga en cuenta lo siguiente:

- z/OS Los mensajes que recibe en z/OS son idénticos a los recibidos en la tarea anterior ya que los mismos objetos de tema y perfiles controlan el acceso.
- Multi El mensaje de suceso que recibe en Multiplatforms es similar al recibido en la tarea anterior, pero la serie de tema real es diferente.

Otorgar acceso a otro usuario para suscribirse sólo al tema más profundamente en el árbol

Este tema es el tercero de una lista de tareas que indica cómo otorgar acceso para suscribirse a los temas por parte de más de un usuario.

Antes de empezar

Este tema utiliza la configuración descrita en “Otorgar acceso a un usuario para suscribirse a un tema más profundamente en el árbol” en la página 503.

Acerca de esta tarea

En “Otorgar acceso a un usuario para suscribirse a un tema más profundamente en el árbol” en la página 503, se ha denegado a USER2 el acceso al tema `"Price/Fruit/Apples"`. Este tema indica cómo otorgar acceso a dicho tema, pero no a otros temas.

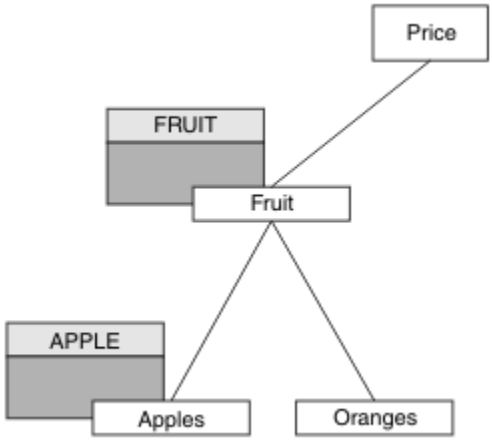


Figura 25. Otorgar acceso a temas específicos dentro de un árbol de temas

Tabla 91. Requisitos de acceso para los temas de ejemplo y los objetos de tema		
Tema	**Acceso de suscripción necesario**	**Objeto de tema**
Price	Ningún usuario	Ninguna
Price/Fruit	USER1	FRUIT
Price/Fruit/Apples	USER1 y USER2	APPLE
Price/Fruit/Oranges	USER1	

Defina un nuevo objeto de tema de la manera siguiente:

Procedimiento

1. Emita el mandato MQSC `DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')`.
2. Otorgue el acceso de la manera siguiente:

- z/OS **z/OS** :

  En “Otorgar acceso a un usuario para suscribirse a un tema más profundamente en el árbol” en la página 503 USER1 se ha otorgado acceso para suscribirse al tema `"Price/Fruit/Apples"` otorgando al usuario acceso al perfil `hlq.SUBSCRIBE.FRUIT` .

  Este perfil único también ha otorgado acceso a USER1 para suscribirse a `"Price/Fruit/Oranges"` `"Price/Fruit/#"` y este acceso permanece incluso con la adición del nuevo objeto de tema y los perfiles asociados al mismo.

  Otorgue acceso a USER2 para suscribirse al tema `"Price/Fruit/Apples"` otorgando acceso de usuario al perfil `hlq.SUBSCRIBE.APPLE`. Para ello, utilice los siguientes mandatos RACF:

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
  PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
  ```

- Multi Multiplataformas:

  En “Otorgar acceso a un usuario para suscribirse a un tema más profundamente en el árbol” en la página 503 USER1 se ha otorgado acceso para suscribirse al tema `"Price/Fruit/Apples"` otorgando al usuario acceso de suscripción al perfil FRUIT .

  Este único perfil también otorgaba acceso a USER1 para suscribirse a `"Price/Fruit/Oranges"` y `"Price/Fruit/#"` y este acceso permanece incluso al agregar el nuevo objeto de tema y los perfiles asociados al mismo.

  Otorgue acceso a USER2 para suscribirse al tema `"Price/Fruit/Apples"` otorgando al usuario acceso de suscripción al perfil APPLE. Hágalo mediante el mandato de autorización para la plataforma:

  ALW **AIX, Linux, and Windows sistemas**

  ```
  setmqaut -t topic -n APPLE -p USER2 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*SUB)
  ```

Resultados

z/OS En z/OS, cuando USER1 intenta suscribirse al tema `"Price/Fruit/Apples"`, la primera comprobación de seguridad en el perfil `hlq.SUBSCRIBE.APPLE` falla, pero al subir en el árbol, el perfil `hlq.SUBSCRIBE.FRUIT` permite que USER1 se suscriba, de forma que la suscripción es satisfactoria y el código se envía a la llamada MQSUB. Sin embargo, se genera un mensaje RACF `ICH` para la primera comprobación:

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

Cuando USER2 intenta suscribirse al tema `"Price/Fruit/Apples"` el resultado es satisfactorio porque la comprobación de seguridad pasa en el primer perfil.

Cuando USER2 intenta suscribirse al tema `"Price/Fruit/Oranges"` el resultado es anómalo con un mensaje `MQRC_NOT_AUTHORIZED`, junto con:

- z/OS En z/OS, se ven los mensajes siguientes en la consola que muestran la vía de acceso de seguridad completa a través del árbol de temas que se ha intentado:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW En plataformas AIX, Linux, and Windows , el siguiente suceso de autorización:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

- IBM i En IBMi, el siguiente suceso de autorización:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

z/OS La desventaja de esta configuración es que, en z/OS, recibirá mensajes `ICH` adicionales en la consola. Puede evitarlo si protege el árbol de temas de forma diferente.

Cambio de control de acceso para evitar mensajes adicionales

Este tema es el cuarto de una lista de tareas que le indica cómo otorgar acceso para suscribirse a temas por más de un usuario y para evitar mensajes adicionales de RACF ICH408I en z/OS.

Antes de empezar

Este tema aumenta la configuración descrita en “Otorgar acceso a otro usuario para suscribirse sólo al tema más profundamente en el árbol” en la página 504 para que evite mensajes de error adicionales.

Acerca de esta tarea

En este tema se describe cómo puede otorgar acceso a los temas más profundos en el árbol y cómo eliminar el acceso al tema más abajo en el árbol cuando ningún usuario lo requiere.

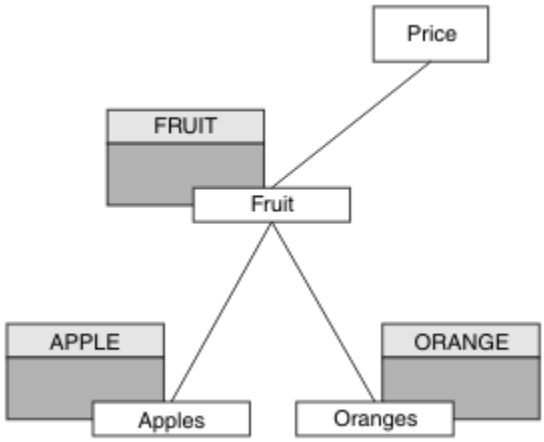


Figura 26. Ejemplo de otorgar control de acceso para evitar mensajes adicionales.

Defina un nuevo objeto de tema de la manera siguiente:

Procedimiento

1. Emita el mandato MQSC `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')`.
2. Otorgue el acceso de la manera siguiente:

- z/OS **z/OS** :

  Defina un nuevo perfil y agregue el acceso a dicho perfil y a los perfiles existentes. Para ello, utilice los siguientes mandatos RACF:

  ```
  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 el acceso equivalente con los mandatos de autorización de la plataforma:

  ALW **AIX, Linux, and Windows sistemas**

  ```
  setmqaut -t topic -n ORANGE -p USER1 +sub
  setmqaut -t topic -n APPLE -p USER1 +sub
  ```

  IBM i **IBM i**

  ```
  GRTMQAUT OBJ(ORANGE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
  ```

Resultados

z/OS En z/OS, cuando USER1 intenta suscribirse al tema `"Price/Fruit/Apples"`, la primera comprobación de seguridad del perfil `hlq.SUBSCRIBE.APPLE` es satisfactoria.

Cuando USER2 intenta suscribirse al tema `"Price/Fruit/Apples"` el resultado es satisfactorio porque la comprobación de seguridad pasa en el primer perfil.

Cuando USER2 intenta suscribirse al tema `"Price/Fruit/Oranges"` el resultado es anómalo con un mensaje `MQRC_NOT_AUTHORIZED`, junto con:

- z/OS En z/OS, se ven los mensajes siguientes en la consola que muestran la vía de acceso de seguridad completa a través del árbol de temas que se ha intentado:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.ORANGE ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW En AIX, Linux, and Windows, el siguiente suceso de autorización:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- IBM i En IBM i, el siguiente suceso de autorización:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

Otorgar acceso a un usuario para publicar en un tema

Este tema es el primero de una lista de tareas que indica cómo otorgar acceso para publicar temas por más de un usuario.

Acerca de esta tarea

En esta tarea se presupone que no existen objetos de temas administrativos en el lado derecho del árbol de temas, ni se han definido los perfiles para la publicación. La suposición utilizada es que los editores usan sólo la serie de tema.

Una aplicación puede publicar en un tema proporcionando un objeto de tema, o una serie de tema, o una combinación de ambos. Sea cual sea lo que seleccione la aplicación, el efecto es publicar en un punto determinado del árbol de temas. Si este punto del árbol de temas está representado por un objeto de tema administrativo, se comprueba un perfil de seguridad según el nombre de este objeto de tema. Por ejemplo:

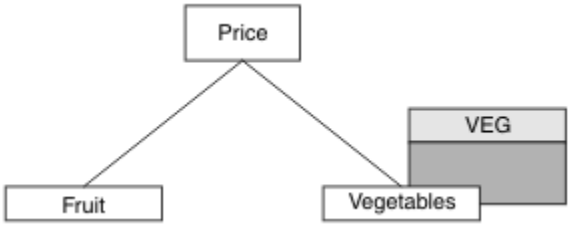


Figura 27. Otorgar acceso de publicación a un tema

Tabla 92. Ejemplo de requisitos de acceso de publicación

Tema	Acceso de publicación necesario	Objeto de tema
Price	Ningún usuario	Ninguna

Tabla 92. Ejemplo de requisitos de acceso de publicación (continuación)		
Tema	**Acceso de publicación necesario**	**Objeto de tema**
Price/Vegetables	USER1	VEG

Defina un nuevo objeto de tema de la manera siguiente:

Procedimiento

1. Emita el mandato MQSC `DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')`.
2. Otorgue el acceso de la manera siguiente:
   - z/OS **z/OS** :

     Otorgue acceso a USER1 para publicar en el tema `"Price/Vegetables"` otorgando acceso de usuario al perfil `hlq.PUBLISH.VEG`. Para ello, utilice los siguientes mandatos RACF:

     ```
     RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
     PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
     ```

   - Otras plataformas:

     Otorgue acceso a USER1 para publicar en el tema `"Price/Vegetables"` otorgando acceso de usuario al perfil VEG. Hágalo mediante el mandato de autorización para la plataforma:

     ALW **AIX, Linux, and Windows sistemas**

     ```
     setmqaut -t topic -n VEG -p USER1 +pub
     ```

     IBM i **IBM i**

     ```
     GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
     ```

Resultados

Cuando USER1 intenta publicar en el tema `"Price/Vegetables"`, el resultado es satisfactorio; es decir, la llamada MQOPEN es satisfactoria.

Cuando USER2 intenta publicar en el tema `"Price/Vegetables"`, el resultado es anómalo con un mensaje `MQRC_NOT_AUTHORIZED`, junto con:

- z/OS En z/OS, se ven los mensajes siguientes en la consola que muestran la vía de acceso de seguridad completa a través del árbol de temas que se ha intentado:

  ```
  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.VEG ...

  ICH408I USER(USER2   ) ...
    hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
  ```

- ALW En otras plataformas, el siguiente suceso de autorización:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
  UserIdentifier        USER2
  ```

```
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

- IBM i En IBMi, el siguiente suceso de autorización:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

Tenga en cuenta que es una ilustración de lo que verá, no de todos los campos.

Otorgar acceso a un usuario para publicar en un tema más profundamente en el árbol

Este tema es el segundo de una lista de tareas que indica cómo otorgar acceso a los temas por más de un usuario.

Antes de empezar

Este tema utiliza la configuración descrita en “Otorgar acceso a un usuario para publicar en un tema” en la página 508.

Acerca de esta tarea

Si el punto del árbol de temas en que la aplicación realiza la publicación no está representada por un objeto de tema administrativo, suba en el árbol hasta localizar el objeto de tema administrativo padre más cercano. El perfil de seguridad se comprueba, basándose en el nombre de dicho objeto de tema.

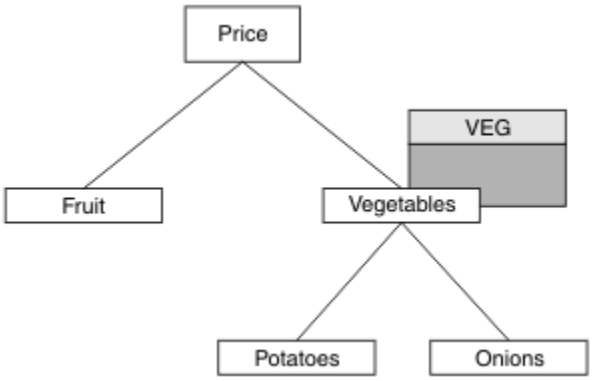


Figura 28. Otorgar acceso de publicación a un tema dentro de un árbol de temas

Tabla 93. Ejemplo de requisitos de acceso de publicación

Tema	Acceso de suscripción necesario	Objeto de tema
Price	Ningún usuario	Ninguna
Price/Vegetables	USER1	VEG
Price/ Vegetables/ Potatoes	USER1	
Price/ Vegetables/ Onions	USER1	

En la tarea anterior, a USER1 se le ha otorgado acceso para publicar el tema "Price/Vegetables/Potatoes" otorgándole acceso al perfil hlq.PUBLISH.VEG en z/OS o acceso de publicación al perfil VEG en Multiplatforms. Este perfil único también otorga acceso a USER1 para publicar en "Price/Vegetables/Onions".

Cuando USER1 intenta publicar en el tema "Price/Vegetables/Potatoes", el resultado es satisfactorio; es decir, la llamada MQOPEN es satisfactoria.

Cuando USER2 intenta suscribirse al tema "Price/Vegetables/Potatoes", el resultado es anómalo; es decir, la llamada MQOPEN falla con un mensaje MQRC_NOT_AUTHORIZED, junto con:

- z/OS En z/OS, se ven los mensajes siguientes en la consola que muestran la vía de acceso de seguridad completa a través del árbol de temas que se ha intentado:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Multi En Multiplatforms, el siguiente suceso de autorización:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

Tenga en cuenta lo siguiente:

- z/OS Los mensajes que recibe en z/OS son idénticos a los recibidos en la tarea anterior ya que los mismos objetos de tema y perfiles controlan el acceso.
- Multi El mensaje de suceso que recibe en Multiplatforms es similar al recibido en la tarea anterior, pero la serie de tema real es diferente.

Otorgar acceso para publicar y suscribir

Este tema es el último de una lista de tareas que indica cómo otorgar acceso para publicar y suscribirse a temas por más de un usuario.

Antes de empezar

Este tema utiliza la configuración descrita en “Otorgar acceso a un usuario para publicar en un tema más profundamente en el árbol” en la página 510.

Acerca de esta tarea

En una tarea anterior se otorgó acceso a USER1 para suscribirse al tema "Price/Fruit". Este tema explica cómo otorgar acceso a dicho usuario para publicar en dicho tema.

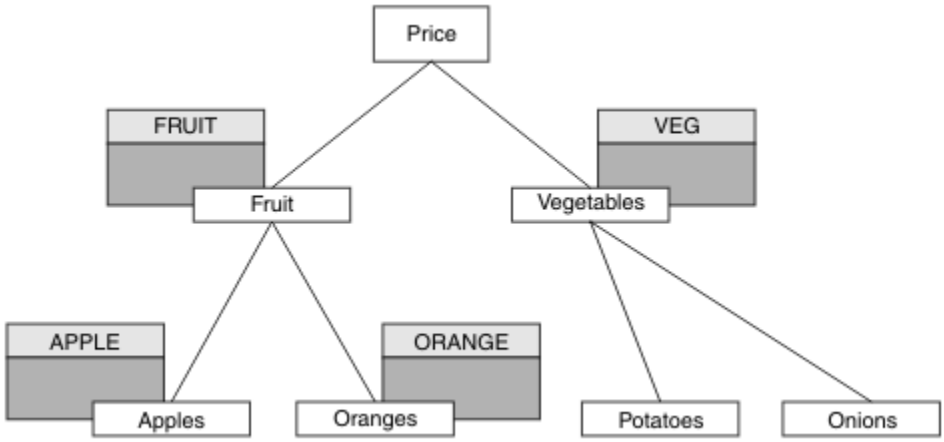


Figura 29. Otorgar acceso para publicar y suscribir

Tabla 94. Ejemplo de requisitos de acceso de publicación y suscripción

Tema	Acceso de suscripción necesario	Acceso de publicación necesario	Objeto de tema
Price	Ningún usuario	Ningún usuario	Ninguna
Price/Fruit	USER1	USER1	FRUIT
Price/Fruit/Apples	USER1 y USER2		APPLE
Price/Fruit/Oranges	USER1		ORANGE

Procedimiento

Otorgue el acceso de la manera siguiente:

- z/OS **z/OS** :

  En una tarea anterior se otorgó acceso a USER1 para suscribirse al tema `"Price/Fruit"` otorgando al usuario acceso al perfil `hlq.SUBSCRIBE.FRUIT`.

  Para publicar en el tema `"Price/Fruit"`, otorgue acceso a USER1 al perfil `hlq.PUBLISH.FRUIT`. Para ello, utilice los siguientes mandatos RACF:

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
  PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Multi Multiplataformas:

  Otorgue acceso a USER1 para publicar en el tema `"Price/Fruit"` otorgando acceso de publicación de usuario al objeto FRUIT. Hágalo mediante el mandato de autorización para la plataforma:

  ALW **AIX, Linux, and Windows sistemas**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +pub
  ```

IBM i

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
```

Resultados

En z/OS, cuando USER1 intenta publicar en el tema `"Price/Fruit"` se pasa la comprobación de seguridad en la llamada MQOPEN.

Cuando USER2 intenta publicar en el tema `"Price/Fruit"` el resultado es anómalo con un mensaje `MQRC_NOT_AUTHORIZED`, junto con:

- En z/OS, se ven los mensajes siguientes en la consola que muestran la vía de acceso de seguridad completa a través del árbol de temas que se ha intentado:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- En plataformas AIX, Linux, and Windows , el siguiente suceso de autorización:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- En IBM i, el siguiente suceso de autorización:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

Tras el conjunto completo de estas tareas, ofrece a USER1 y USER2 las autorizaciones de acceso siguientes para publicar y suscribirse a los temas que se listan:

Tabla 95. Lista completa de las autorizaciones de acceso resultantes de ejemplos de seguridad

Tema	Acceso de suscripción necesario	Acceso de publicación necesario	Objeto de tema
Price	Ningún usuario	Ningún usuario	Ninguna
Price/Fruit	USER1	USER1	FRUIT
Price/Fruit/Apples	USER1 y USER2		APPLE
Price/Fruit/Oranges	USER1		ORANGE
Price/Vegetables		USER1	VEG

Tabla 95. Lista completa de las autorizaciones de acceso resultantes de ejemplos de seguridad (continuación)

Tema	Acceso de suscripción necesario	Acceso de publicación necesario	Objeto de tema
Price/ Vegetables/ Potatoes			
Price/ Vegetables/ Onions			

z/OS Donde tenga requisitos distintos de acceso de seguridad a distintos niveles dentro del árbol de temas, una planificación cuidadosa asegura que no recibirá avisos de seguridad improcedentes en el registro de la consola de z/OS. La configuración de seguridad en el nivel correcto dentro del árbol evita mensajes de seguridad engañosos.

Seguridad de suscripción

MQSO_ALTERNATE_USER_AUTHORITY

El campo AlternateUserId contiene un identificador de usuario para utilizarlo para validar esta llamada MQSUB. La llamada solo será satisfactoria si este AlternateUserId tiene autorización para suscribirse al tema con las opciones de acceso especificadas, independientemente de si el identificador del usuario bajo el que está ejecutándose la aplicación tiene autorización para ello.

MQSO_SET_IDENTITY_CONTEXT

La suscripción debe utilizar la señal de contabilidad y los datos de identidad de la aplicación suministrados en los campos PubAccountingToken y PubApplIdentityData.

Si se especifica esta opción, se realiza la misma comprobación de autorización que si se accediera a la cola de destino mediante una llamada MQOPEN con MQOO_SET_IDENTITY_CONTEXT, excepto en el caso en que se utilice también la opción MQSO_MANAGED, en cuyo caso no hay comprobación de autorización en la cola de destino.

Si no se especifica esta opción, las publicaciones enviadas a este suscriptor tienen información de contexto predeterminada asociada a ellos, de la manera siguiente:

Tabla 96. Información de contexto de publicación predeterminado

Campo de MQMD	Valor utilizado
UserIdentifier	El ID de usuario asociado a la suscripción (vea el campo SUBUSER en DISPLAY SBSTATUS) cuando se realizó la publicación.
AccountingToken	Determinado por el entorno si es posible; en caso contrario, establecido en MQACT.
ApplIdentityData	Establecido en blancos.

Esta opción sólo es válida con MQSO_CREATE y MQSO_ALTER. Si se utiliza con MQSO_RESUME, los campos PubAccountingToken y PubApplIdentityData se ignoran, por lo que esta opción no tiene ningún efecto.

Si se modifica una suscripción sin utilizar esta opción en la que previamente la suscripción había facilitado información de contexto de identidad, se genera información del contexto predeterminado para la suscripción modificada.

Si una suscripción que permite que distintos ID de usuario la utilicen con la opción MQSO_ANY_USERID, se reanuda con un ID de usuario diferente, se genera contexto de identidad predeterminado para el nuevo ID de usuario que es propietario ahora de la suscripción y las publicaciones posteriores se entregan conteniendo el nuevo contexto de identidad.

AlternateSecurityId

Este es un identificador de seguridad que se transfiere con el AlternateUserId al servicio de autorizaciones para permitir que se realicen las comprobaciones de autorización correspondientes. AlternateSecurityId sólo se utiliza si se especifica MQSO_ALTERNATE_USER_AUTHORITY y el campo AlternateUserId no está completamente en blanco hasta el primer carácter nulo o el final del campo.

Opción de suscripción MQSO_ANY_USERID

Cuando se especifica MQSO_ANY_USERID, la identidad del suscriptor no está restringida a un ID de usuario único. Esto permite que cualquier usuario modifique o reanude la suscripción cuando disponga de la autoridad adecuada. Sólo puede tener la suscripción un único usuario a la vez. Un intento de reanudar el uso de una suscripción utilizada actualmente por otra aplicación hará que falle la llamada con MQRC_SUBSCRIPTION_IN_USE.

Para añadir esta opción a una suscripción existente, la llamada MQSUB (utilizando MQSO_ALTER) debe proceder del mismo ID de usuario que la suscripción original.

Si una llamada MQSUB hace referencia a una suscripción existente con MSQO_ANY_USERID establecido y el ID de usuario difiere de la suscripción original, la llamada sólo será satisfactoria si el nuevo ID de usuario tiene autorización para suscribirse al tema. Tras la finalización satisfactoria, las futuras publicaciones de este suscriptor se colocarán en la cola del suscriptor con el nuevo ID de usuario establecido en la publicación.

MQSO_FIXED_USERID

Cuando se especifica MQSO_FIXED_USERID, sólo un ID de usuario propietario puede modificar o reanudar la suscripción. Este ID de usuario es el último ID de usuario para modificar la suscripción que estableció esta opción, eliminando así la opción MQSO_ANY_USERID, o si se ha llevado a cabo ninguna modificación, es el ID de usuario que ha creado la suscripción.

Si un verbo MQSUB hace referencia a una suscripción existente con MQSO_ANY_USERID establecido y modifica la suscripción (utilizando MQSO_ALTER) para utilizar la opción MQSO_FIXED_USERID, el ID de usuario de la suscripción se ha fijado ahora en este ID de usuario nuevo. La llamada sólo es satisfactoria si el nuevo ID de usuario tiene autoridad para suscribirse al tema.

Si un ID de usuario distinto del registrado como propietario de una suscripción intenta reanudar o modificar una suscripción MQSO_FIXED_USERID, la llamada fallará con MQRC_IDENTITY_MISMATCH. El ID de usuario propietario de una suscripción se puede ver mediante el mandato DISPLAY SBSTATUS.

Si no se especifica MQSO_ANY_USERID ni MQSO_FIXED_USERID, el valor predeterminado es MQSO_FIXED_USERID.

Seguridad de publicación/suscripción entre gestores de colas

Los mensajes internos de publicación/suscripción como, por ejemplo, las suscripciones y publicaciones proxy, se colocan en colas de sistema de publicación/suscripción mediante las reglas normales de seguridad de canal. La información y diagramas de este tema resaltan los diversos procesos y los ID de usuario implicados en la entrega de estos mensajes.

Control de acceso local

El acceso a temas para publicación y suscripciones se rige por las definiciones de seguridad locales y las reglas que se describen en Seguridad de publicación/suscripción. No es necesario ningún objeto de tema local para establecer el control de acceso. Los administradores pueden optar por aplicar el control de acceso a objetos de tema de clúster, independientemente de si existen en el clúster todavía.

Los administradores de sistema son responsables del control de acceso en su sistema local. Deben confiar en los administradores de otros miembros de la jerarquía o colectivos de clúster de ser responsables de su política de control de acceso. Como el control de acceso está definido para cada máquina por separado, es probable que sea un contratiempo si se necesita un control de nivel muy preciso. Puede que no sea necesario imponer ningún control de acceso, o que pueda definirse el control de acceso en los objetos de alto nivel del árbol de temas. Puede definirse un control de acceso de nivel más preciso para cada subdivisión del espacio de nombres de temas.

Realizar una suscripción proxy

La confianza de una organización al conectar su gestor de colas al gestor de colas del usuario se confirma por medios normales de autenticación de canal. Si a la organización de confianza también se le permite realizar una publicación/suscripción distribuida, se efectúa una comprobación de autoridad. La comprobación se realiza cuando el canal coloca un mensaje en una cola de publicación/suscripción distribuida. Por ejemplo, si se coloca un mensaje en la cola `SYSTEM.INTER.QMGR.CONTROL` . El ID de usuario para la comprobación de autorización de cola depende de los valores de PUTAUT del canal receptor. Por ejemplo, el ID de usuario del canal, `MCAUSER`, el contexto del mensaje, según el valor y plataforma. Para obtener más información sobre la seguridad de canal, consulte Seguridad de canal.

Las suscripciones proxy se realizan con el ID de usuario del agente de publicación/suscripción distribuido en el gestor de colas remoto. Por ejemplo, `QM2` en Figura 30 en la página 516. Entonces se otorga acceso al usuario fácilmente a los perfiles de objeto de tema locales, porque dicho ID de usuario está definido en el sistema y, por consiguiente, no hay conflictos de dominio.

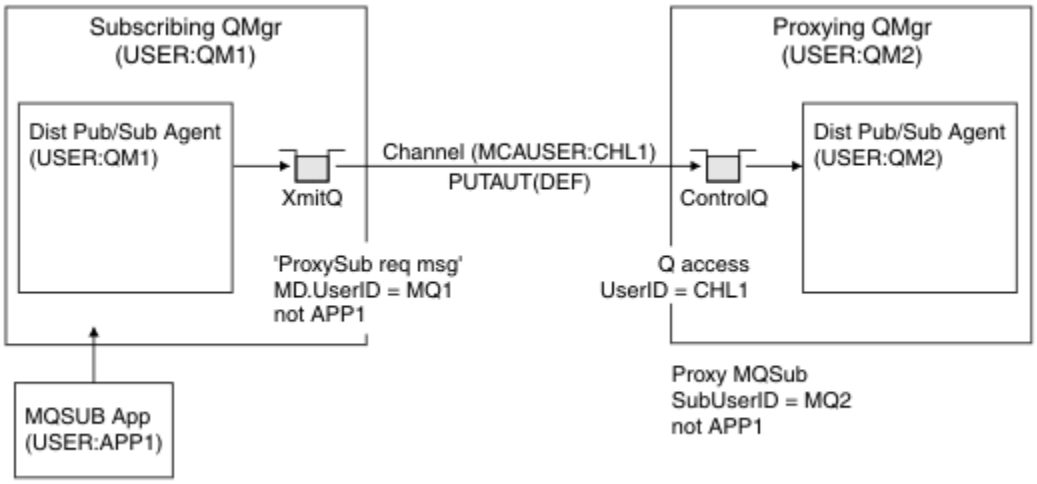


Figura 30. Seguridad de suscripción proxy, realizar una suscripción

Devolver publicaciones remotas

Cuando se crea una publicación en el gestor de colas de publicación, una copia de la publicación se crea para cualquier suscripción proxy. El contexto de la publicación copiada contiene el contexto del ID de usuario que hizo la suscripción; `QM2` en Figura 31 en la página 517. La suscripción proxy se crea con una cola de destino que es una cola remota, por lo que el mensaje de publicación se resuelve en una cola de transmisión.

De confianza para una organización al conectar su gestor de colas, `QM2`, a otro gestor de colas, `QM1`, se confirma por medios normales de autenticación de canal. Si se permite a la organización de confianza

realizar la publicación/suscripción distribuida, se lleva a cabo una comprobación de autorización cuando el canal coloca el mensaje de publicación en la cola de publicación de la publicación/suscripción distribuida SYSTEM.INTER.QMGR.PUBS. El ID de usuario de la comprobación de autorización de cola depende del valor de PUTAUT del canal receptor (por ejemplo, el ID de usuario del canal, MCAUSER, contexto de mensaje y otros, en función del valor y la plataforma). Para obtener más información sobre la seguridad de canal, consulte Seguridad de canal.

Cuando el mensaje de publicación llega al gestor de colas de suscripción, se realiza otro MQPUT en el tema bajo la autorización de dicho gestor de colas y el contexto que incluye el mensaje es sustituido por el contexto de cada uno de los suscriptores locales a medida que se les entrega el mensaje.

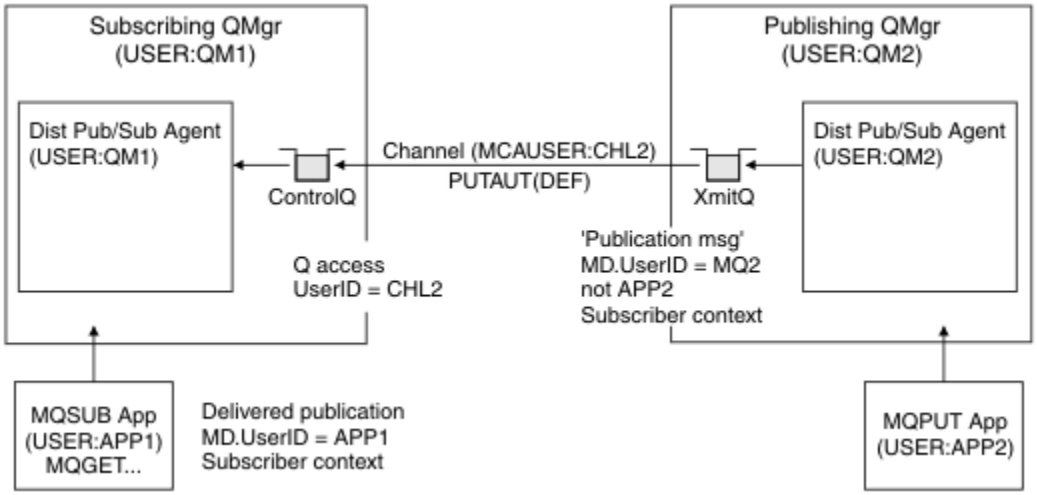


Figura 31. Seguridad de suscripción proxy, reenviando publicaciones

En un sistema en el que poco se ha tenido en cuenta en relación con la seguridad, los procesos de publicación/suscripción distribuidos probablemente se estén ejecutando bajo un ID de usuario del grupo mqm, el parámetro MCAUSER en un canal está en blanco (el valor predeterminado), y los mensajes se entregan a las diversas colas del sistema, según sea necesario. El sistema no seguro facilita la configuración de una prueba de concepto para demostrar la publicación/suscripción distribuida.

En un sistema donde la seguridad es considerada más seriamente, estos mensajes internos están sujetos a los controles de seguridad igual que cualquier mensaje que pase a través del canal.

Si el canal está configurado con un carácter no blanco MCAUSER y un valor PUTAUT que especifica que MCAUSER debe comprobarse, entonces debe concederse al MCAUSER en cuestión acceso a las colas SYSTEM.INTER.QMGR.*. Si hay varios gestores de colas remotos diferentes, con canales que se ejecutan con distintos ID de MCAUSER, es necesario otorgar a todos los ID de usuario acceso a las colas SYSTEM.INTER.QMGR.*. Pueden aparecer canales que se ejecutan con ID de MCAUSER diferentes cuando, por ejemplo, varias conexiones jerárquicas se configuran en un único gestor de colas.

Si el canal está configurado con un valor PUTAUT que especifica que se utiliza el contexto del mensaje, entonces el acceso a la colas SYSTEM.INTER.QMGR.* se comprueba basándose en el ID de usuario dentro del mensaje interno. Dado que todos estos mensajes se transfieren con el ID de usuario del agente de publicación/suscripción distribuido del gestor de colas que envía el mensaje interno o el mensaje de publicación (consulte Figura 31 en la página 517), un conjunto de ID de usuario para otorgar acceso a las diversas colas de sistema no es demasiado grande (uno por cada gestor de colas remoto), si desea configurar la seguridad de publicación/suscripción distribuida de esta manera. Aún tiene las mismas cuestiones que siempre tiene la seguridad de contexto de canal; las de los diferentes dominios de ID de usuario y el hecho de que el ID de usuario en el mensaje podría no estar definido en el sistema receptor. Sin embargo, es una manera perfectamente aceptable de ejecutarlo si es necesario.

z/OS La sección Seguridad de colas del sistema proporciona una lista de colas y el acceso que se necesita para configurar de forma segura el entorno de publicación/suscripción distribuida. Si los mensajes internos o publicaciones no se transfieren debido a violaciones de seguridad, el canal escribe

un mensaje en el registro de la forma normal y los mensajes se pueden enviar a la cola de mensajes no entregados de acuerdo con el proceso de errores de canal normal.

Todos los mensajes entre gestores de colas a efectos de publicación/suscripción distribuida se ejecutan utilizando la seguridad de canal normal.

Para obtener información sobre la restricción de las publicaciones y suscripciones proxy en el nivel de tema, consulte Seguridad de publicación/suscripción.

Utilización de los ID de usuario predeterminados con una jerarquía de gestores de colas

Si tiene una jerarquía de gestores de colas que se ejecutan en plataformas diferentes y utilizan los ID de usuario predeterminados, tenga en cuenta que estos ID de usuario predeterminados difieren entre plataformas y es posible que no sean conocidos en la plataforma de destino. Como resultado, un gestor de colas que se ejecuta en una plataforma rechaza los mensajes recibidos de los gestores de colas de otras plataformas con el código de razón `MQRC_NOT_AUTHORIZED`.

Para evitar que se rechacen mensajes, como mínimo, las autorizaciones siguientes deben añadirse a los ID de usuario predeterminados utilizados en otras plataformas:

- Autorización *PUT *GET en las colas SYSTEM.BROKER. colas
- Autorización *PUB *SUB en los temas SYSTEM.BROKER. temas
- Autorización *ADMCRT *ADMDLT *ADMCHG en la cola SYSTEM.BROKER.CONTROL.QUEUE.

Los ID de usuario predeterminados con una jerarquía de gestores de colas son los siguientes:

Plataforma	**ID de usuario predeterminado**
Windows	`mqm`
Sistemas AIX and Linux	`mqm`
IBM i	`QMQM`
z/OS	El ID de usuario del espacio de direcciones del iniciador de canal

Si los gestores de colas en plataformas distintas de IBM i están conectados jerárquicamente a un gestor de colas en IBM i, cree y otorgue acceso al ID de usuario 'qmqm'.

Si los gestores de colas en IBM i o z/OS están conectados jerárquicamente a un gestor de colas en AIX, Linux, and Windows, cree y otorgue acceso al ID de usuario 'mqm'.

Si los gestores de colas en Multiplatforms están conectados jerárquicamente a un gestor de colas en z/OS, cree y otorgue acceso al ID de usuario del espacio de direcciones del iniciador de canal z/OS .

Los ID de usuario pueden distinguir entre mayúsculas y minúsculas. El gestor de colas de origen (si está en Multiplatforms) fuerza que el ID de usuario esté todo en mayúsculas. El gestor de colas receptor (si está en AIX, Linux, and Windows) fuerza que el ID de usuario esté todo en minúsculas. Por lo tanto, todos los ID de usuario creados en los sistemas AIX and Linux deben crearse en minúsculas. Si se ha instalado una salida de mensaje, no se fuerza al ID de usuario a escribirse en mayúsculas o minúsculas. Hay que tener cuidado para comprender cómo la salida de mensajes procesa el ID de usuario.

Para evitar posibles problemas con la conversión de los ID de usuario:

- En los sistemas AIX, Linux, and Windows, asegúrese de que los ID de usuario se han especificado en minúsculas.
- En sistemas IBM i y z/OS , asegúrese de que los ID de usuario se especifiquen en mayúsculas.

Seguridad de IBM MQ Console y REST API

La seguridad de IBM MQ Console y REST API se configura añadiendo la configuración del servidor mqweb en el archivo `mqwebuser.xml`.

Acerca de esta tarea

Puede realizar un seguimiento de las acciones de usuario y auditar el uso de IBM MQ Console y REST API examinando los archivos de registro del servidor mqweb.

Los usuarios de IBM MQ Console y REST API se pueden autenticar utilizando:

- Un registro básico
- Un registro LDAP
- Un registro de sistema operativo
- SAF en z/OS
- Cualquier otro tipo de registro soportado por WebSphere Liberty

Los roles se pueden asignar a usuarios IBM MQ Console y a usuarios REST API para determinar el nivel de acceso que se les otorga a los objetos de IBM MQ. Por ejemplo, para realizar la mensajería, los usuarios deben tener asignado el rol `MQWebUser`. Si desea más información sobre los roles disponibles, consulte "Roles en IBM MQ Console y REST API" en la página 531.

Una vez asignado un rol a un usuario, pueden utilizarse varios métodos para autenticar el usuario. Con IBM MQ Console, los usuarios pueden iniciar una sesión con un nombre de usuario y una contraseña, o pueden utilizar la autenticación de certificados de cliente. Con la REST API, los usuarios pueden utilizar la autenticación HTTP básica, la autenticación basada en señal o la autenticación de certificado de cliente.

Procedimiento

1. Defina el registro de usuarios para autenticar los usuarios y asigne a cada usuario o grupo un rol para que puedan utilizar IBM MQ Console o REST API. Para obtener más información, consulte "Configuración de usuarios y roles" en la página 520
2. Elija cómo se autentican los usuarios de IBM MQ Console con el servidor mqweb. No es necesario que utilice el mismo método para todos los usuarios:
   - Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario especifica un ID de usuario y una contraseña en la pantalla de inicio de sesión de IBM MQ Console. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. No se requiere ninguna configuración adicional para utilizar esta opción de autenticación, pero puede configurar de manera opcional la hora de caducidad de la señal LTPA. Si desea más información, consulte Configuración del intervalo de caducidad de señal LTPA.
   - Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en IBM MQ Console, sino que utiliza el certificado de cliente. Para obtener más información, consulte "Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console" en la página 535.
3. Elija cómo se autentican los usuarios de REST API con el servidor mqweb. No es necesario que utilice el mismo método para todos los usuarios:
   - Los usuarios se autentican utilizando la autenticación básica HTTP. En este caso, se codifican un nombre de usuario y una contraseña, pero no se cifran, y se envían a cada solicitud REST API para autenticar y autorizar al usuario para esa solicitud. Para que esta autenticación sea segura, debe utilizar una conexión segura. Es decir, debe utilizar HTTPS. Para obtener más información, consulte "Utilización de la autenticación básica HTTP con REST API" en la página 538.
   - Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario proporciona un ID de usuario y una contraseña al recurso REST API `login` con el método HTTP POST. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. Para obtener más información, consulte "Utilización de la autenticación basada en señal con la API REST" en la página 539.

     Para que esta autenticación sea segura, debe utilizar una conexión segura. Es decir, debe utilizar HTTPS. Sin embargo, si ha habilitado las conexiones HTTP, puede permitir que se emita una señal LTPA para que se utilice una conexión HTTPS para una conexión HTTP. Para obtener más información, consulte Configuración de la señal LTPA.

- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en REST API, sino que utiliza el certificado de cliente. Para obtener más información, consulte "Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console" en la página 535.

4. Opcional: Configure Cross Origin Resource Sharing (CORS) para REST API.

   De forma predeterminada, un navegador web no permite que scripts como, por ejemplo, JavaScript, invoquen REST API cuando no provienen del mismo origen que REST API. Es decir, las solicitudes entre orígenes no están habilitadas. Puede configurar Cross Origin Resource Sharing (CORS) para permitir las solicitudes entre orígenes a partir de los URL especificados. Para obtener más información, consulte "Configuración de CORS para REST API" en la página 542.

5. Opcional: Configure la validación de la cabecera de host para la IBM MQ Console y la REST API.

   Puede configurar la validación de cabecera de host y crear una lista de elementos permitidos de nombres de host y puertos para asegurarse de que IBM MQ Console y REST APIsólo procesan las solicitudes que contienen cabeceras de host específicas. Para obtener más información, consulte "Configurando la validación de la cabecera de host para la IBM MQ Console y la REST API" en la página 543.

Configuración de usuarios y roles

Para poder utilizar IBM MQ Console o REST API, los usuarios deben autenticarse en un registro de usuarios, definido en el servidor mqweb.

Acerca de esta tarea

Los usuarios autenticados deben ser miembros de uno de los grupos que autoriza el acceso a las prestaciones de IBM MQ Console y REST API. De forma predeterminada, el registro de usuarios no contiene ningún usuario; estos deben añadirse editando el archivo `mqwebuser.xml`.

Cuando se configuran usuarios y grupos, primero debe configurar un registro de usuarios con el que autenticar los usuarios y los grupos. Este registro de usuarios está compartido entre la IBM MQ Console y la REST API. Puede controlar si los usuarios y grupos tienen acceso a IBM MQ Console, REST API, o a ambos, mediante la configuración de roles para los usuarios y grupos.

Después de configurar el registro de usuarios, puede configurar roles para los usuarios y grupos para otorgarles autorizaciones. Existen varios roles disponibles, incluyendo roles específicos al uso de REST API para Managed File Transfer. Cada rol otorga un nivel de acceso diferente. Para obtener más información, consulte "Roles en IBM MQ Console y REST API" en la página 531.

Se proporcionan varios archivos XML de ejemplo con el servidor mqweb para simplificar la configuración de usuarios y grupos. Es posible que los usuarios que están familiarizados con la configuración de la seguridad en WebSphere Liberty (WLP) prefieran no utilizar los ejemplos. WLP proporciona otras funciones de autorización además de las documentadas aquí.

Procedimiento

- Configure usuarios y grupos con un registro básico utilizando el archivo `basic_registry.xml`.

  Los nombres de usuario y las contraseñas en el registro se utilizan para autenticar y autorizar usuarios de IBM MQ Console y REST API.

  Para configurar un registro básico utilizando el archivo de ejemplo `basic_registry.xml`, consulte "Configuración de un registro básico para IBM MQ Console y REST API" en la página 522.

- Configure usuarios y grupos con un registro LDAP utilizando el archivo `ldap_registry.xml`.

  Los nombres de usuario y las contraseñas en el registro LDAP se utilizan para autenticar y autorizar usuarios de IBM MQ Console y REST API.

  Para configurar un registro LDAP utilizando el archivo de ejemplo `ldap_registry.xml`, consulte "Configuración de un registro LDAP para la IBM MQ Console y la REST API" en la página 526.

- ALW
  Configure usuarios y grupos con un registro de sistema operativo local utilizando el archivo `local_os_registry.xml`.

  Los nombres de usuario y las contraseñas en el registro del sistema operativo se utilizan para autenticar y autorizar usuarios de IBM MQ Console y REST API.

  Para configurar un registro de SO local utilizando el archivo de ejemplo `local_os_registry.xml`, consulte "Configuración de un registro de SO local para la IBM MQ Console y la REST API" en la página 525.
- z/OS
  Configure usuarios y grupos con la interfaz SAF (System Authorization Facility) en z/OS utilizando el archivo `zos_saf_registry.xml`.

  Los perfiles RACF, u otro producto de seguridad, se utilizan para otorgar a los usuarios y grupos acceso a los roles. Los nombres de usuario y las contraseñas en la base de datos RACF se utilizan para autenticar y autorizar usuarios de IBM MQ Console y REST API.

  Para configurar la interfaz SAF utilizando el archivo de ejemplo `zos_saf_registry.xml`, consulte "Configuring a SAF registry for the IBM MQ Console and REST API" en la página 528.
- Inhabilite la seguridad, incluyendo la capacidad de acceder a la IBM MQ Console, o a la REST API, utilizando el archivo `no_security.xml`.

Qué hacer a continuación

Elija cómo se autentican los usuarios:

Opciones de autenticación de IBM MQ Console

- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario especifica un ID de usuario y una contraseña en la pantalla de inicio de sesión de IBM MQ Console. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. No se requiere ninguna configuración adicional para utilizar esta opción de autenticación, pero puede configurar de manera opcional el intervalo de caducidad de la señal LTPA. Si desea más información, consulte Configuración del intervalo de caducidad de señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en IBM MQ Console, sino que utiliza el certificado de cliente. Para obtener más información, consulte "Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console" en la página 535.

Opciones de autenticación de REST API

- Los usuarios se autentican utilizando la autenticación básica HTTP. En este caso, se codifican un nombre de usuario y una contraseña, pero no se cifran, y se envían a cada solicitud REST API para autenticar y autorizar al usuario para esa solicitud. Para que esta autenticación sea segura, debe utilizar una conexión segura. Es decir, debe utilizar HTTPS. Para obtener más información, consulte "Utilización de la autenticación básica HTTP con REST API" en la página 538.
- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario proporciona un ID de usuario y una contraseña al recurso REST API `login` con el método HTTP POST. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. Para obtener más información, consulte "Utilización de la autenticación basada en señal con la API REST" en la página 539. Puede configurar el intervalo de caducidad de la señal LTPA. Para obtener más información, consulte Configuración de la señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en REST API, sino que utiliza el certificado de cliente. Para obtener más información, consulte "Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console" en la página 535.

Configuración de un registro básico para IBM MQ Console y REST API

Puede configurar un registro básico dentro del archivo `mqwebuser.xml`. Los nombres de usuario, contraseñas y roles del archivo xml se utiliza para autenticar y autorizar usuarios de IBM MQ Console y REST API.

Antes de empezar

- Al configurar usuarios dentro del registro básico, debe asignar un rol a cada usuario. Cada rol proporciona distintos niveles de privilegio para acceder a IBM MQ Console y REST API, y determina el contexto de seguridad que se utiliza cuando se intenta una operación permitida. Tendrá que comprender estos roles antes de poder configurar el registro básico. Si desea más información sobre cada uno de los roles, consulte "Roles en IBM MQ Console y REST API" en la página 531.
- Para completar esta tarea, debe ser un usuario con privilegios suficientes para editar el archivo `mqwebuser.xml`:
  - z/OS En z/OS, debe tener acceso de escritura en el archivo `mqwebuser.xml`.
  - Multi En todos los demás sistemas operativos, debe ser un usuario con privilegios.
  - Linux V 9.4.0 Si el servidor mqweb forma parte de una instalación autónoma de IBM MQ Web Server , debe tener acceso de escritura al archivo `mqwebuser.xml` en el directorio de datos IBM MQ Web Server .

Procedimiento

1. Copie el archivo XML de ejemplo `basic_registry.xml` de una de las vías de acceso siguientes:
   - En una instalación de IBM MQ :
     - ALW En AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS En z/OS: `PathPrefix /web/mq/samp/configuration`

       donde `PathPrefix` es la vía de acceso de instalación de IBM MQ for z/OS UNIX System Services Components .
   - Linux V 9.4.0 En una instalación autónoma de IBM MQ Web Server : *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     donde *MQWEB_INSTALLATION_PATH* es el directorio en el que se ha descomprimido el archivo de instalación de IBM MQ Web Server .
2. Coloque el archivo de ejemplo en el directorio adecuado:
   - En una instalación de IBM MQ :
     - Linux AIX En AIX o Linux: `/var/mqm/web/installations/`*installationName*`/servers/mqweb`
     - Windows En Windows: *MQ_DATA_PATH*`\web\installations\`*installationName*`\servers\mqweb`, donde *MQ_DATA_PATH* es la vía de acceso de datos de IBM MQ . Esta vía de acceso es la vía de acceso de datos que se selecciona durante la instalación de IBM MQ. De forma predeterminada, esta vía de acceso es `C:\ProgramData\IBM\MQ`.
     - z/OS En z/OS: *WLP_user_directory*`/servers/mqweb`

       donde *directorio_usuario_WLP* es el directorio que se ha especificado cuando se ejecutó el script **crtmqweb** para crear la definición del servidor mqweb.

- Linux V 9.4.0 En una instalación autónoma de IBM MQ Web Server : *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

  donde *MQ_OVERRIDE_DATA_PATH* es el directorio de datos de IBM MQ Web Server al que apunta la variable de entorno **MQ_OVERRIDE_DATA_PATH** .

3. Opcional: Si ha cambiado valores de configuración en `mqwebuser.xml`, cópielos en el archivo de ejemplo.
4. Suprima el archivo `mqwebuser.xml` existente y cambie el nombre del archivo de ejemplo a `mqwebuser.xml`.
5. Edite el nuevo archivo `mqwebuser.xml` para añadir usuarios y grupos dentro de los códigos **basicRegistry**.

   Tenga en cuenta que cualquier usuario con el rol `MQWebUser` solo puede realizar las operaciones otorgadas al ID de usuario en el gestor de colas. Por tanto, el ID de usuario definido en el registro ha de tener un ID de usuario idéntico en el sistema en el que está instalado IBM MQ. Estos ID de usuario tienen que coincidir en mayúsculas y minúsculas o la correlación entre ellos puede fallar.

   Si desea más información sobre cómo configurar registros de usuarios básicos, consulte Configuración de un registro de usuarios básicos para Liberty en la documentación de WebSphere Liberty.
6. Asigne roles a usuarios y grupos editando el archivo `mqwebuser.xml`:

   Existen varios roles disponibles que autorizan a usuarios y grupos a utilizar IBM MQ Console, y REST API. Cada rol otorga un nivel de acceso diferente. Para obtener más información, consulte “Roles en IBM MQ Console y REST API” en la página 531.

   - Para asignar roles y otorgar acceso a IBM MQ Console, añada los usuarios y los grupos entre las etiquetas **security-role** adecuadas en las etiquetas **<enterpriseApplication id="com.ibm.mq.console">**.
   - Para asignar roles y otorgar acceso a REST API, añada los usuarios y los grupos entre las etiquetas **security-role** adecuadas en las etiquetas **<enterpriseApplication id="com.ibm.mq.rest">**.

   Para obtener ayuda con el formato de la información de usuarios y grupos dentro de las etiquetas **security-role**, consulte los ejemplos.
7. Si ha proporcionado las contraseñas de los usuarios en `mqwebuser.xml`, debería codificar dichas contraseñas para hacerlas más seguras con el mandato **securityUtility encoding** proporcionado por WebSphere Liberty. Si desea más información, consulte Liberty: Mandato securityUtility en la documentación del producto WebSphere Liberty.

Ejemplo

En el siguiente ejemplo, se otorga acceso al grupo `MQWebAdminGroup` a IBM MQ Console con el rol `MQWebAdmin`. Al usuario `reader` se le otorga acceso con el rol `MQWebAdminRO` y al usuario `guest` se le otorga acceso con el rol `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>
```

En el siguiente ejemplo, a los usuarios `reader` y `guest` se les otorga acceso a IBM MQ Console. Al usuario `user` se le otorga acceso a REST API, y a cualquier usuario del grupo `MQAdmin` se le otorga acceso a IBM MQ Console y REST API. Al usuario `mftadmin` se le otorga acceso a 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>
```

Qué hacer a continuación

Elija cómo se autentican los usuarios:

Opciones de autenticación de IBM MQ Console

- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario especifica un ID de usuario y una contraseña en la pantalla de inicio de sesión de IBM MQ Console. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. No se requiere ninguna configuración adicional para utilizar esta opción de autenticación, pero puede configurar de manera opcional el intervalo de caducidad de la señal LTPA. Si desea más información, consulte Configuración del intervalo de caducidad de señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en IBM MQ Console, sino que utiliza el certificado de cliente. Para obtener más información, consulte “Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console” en la página 535.

Opciones de autenticación de REST API

- Los usuarios se autentican utilizando la autenticación básica HTTP. En este caso, se codifican un nombre de usuario y una contraseña, pero no se cifran, y se envían a cada solicitud REST API para autenticar y autorizar al usuario para esa solicitud. Para que esta autenticación sea segura, debe utilizar una conexión segura. Es decir, debe utilizar HTTPS. Para obtener más información, consulte “Utilización de la autenticación básica HTTP con REST API” en la página 538.
- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario proporciona un ID de usuario y una contraseña al recurso REST API `login` con el método HTTP POST. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. Para obtener más información, consulte “Utilización de la autenticación basada en señal con la API REST” en la página 539. Puede configurar el intervalo de caducidad de la señal LTPA. Para obtener más información, consulte Configuración de la señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en REST API, sino que utiliza el certificado de cliente. Para obtener más información, consulte “Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console” en la página 535.

ALW Configuración de un registro de SO local para la IBM MQ Console y la REST API

Puede configurar un registro de sistema operativo local en el archivo `mqwebuser.xml`. Los nombres de usuario y las contraseñas en el sistema operativo local se utilizan para autenticar y autorizar a los usuarios de la IBM MQ Console y la REST API.

Antes de empezar

- En la autenticación de certificados de cliente con la función de autenticación de sistema operativo local, la identidad de usuario es el nombre común (CN) del nombre distinguido (DN) del certificado de cliente. Si la identidad de usuario no existe como usuario del sistema operativo, el inicio de sesión de certificado de cliente fallará y recurrirá a una autenticación basada en contraseña.
- Para completar esta tarea, debe ser un usuario con privilegios suficientes para editar el archivo `mqwebuser.xml`:
  - Linux V 9.4.0 Si el servidor mqweb forma parte de una instalación autónoma de IBM MQ Web Server , debe tener acceso de escritura al archivo `mqwebuser.xml` en el directorio de datos IBM MQ Web Server .
  - Si el servidor mqweb forma parte de una instalación de IBM MQ , debe ser un usuario privilegiado.

Acerca de esta tarea

Con un registro del sistema operativo local, se asigna automáticamente un rol a los usuarios y los grupos.

- A cualquier usuario que forme parte del grupo 'mqm' o del grupo 'QMQMADM' en IBM i, se le otorgan los roles `MQWebAdmin` y `MFTWebAdmin` .
- A todos los demás usuarios se les otorga el rol `MQWebUser` .

Para obtener más información sobre estos roles, consulte “Roles en IBM MQ Console y REST API” en la página 531.

Solo se puede utilizar un registro de sistema operativo local en AIX, Linux, and Windows. z/OS Se proporciona una función equivalente en z/OS configurando un registro SAF. Para obtener más información, consulte “Configuring a SAF registry for the IBM MQ Console and REST API” en la página 528.

Procedimiento

1. Copie el archivo XML de ejemplo `local_os_registry.xml` de una de las vías de acceso siguientes:
   - Linux V 9.4.0 En una instalación autónoma de IBM MQ Web Server : *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     donde *MQWEB_INSTALLATION_PATH* es el directorio en el que se ha descomprimido el archivo de instalación de IBM MQ Web Server .
   - En una instalación de IBM MQ : *MQ_INSTALLATION_PATH*`/web/mq/samp/configuration`
2. Coloque el archivo de ejemplo en uno de los directorios siguientes:
   - Linux V 9.4.0 En una instalación autónoma de IBM MQ Web Server : *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     donde *MQ_OVERRIDE_DATA_PATH* es el directorio de datos de IBM MQ Web Server al que apunta la variable de entorno **MQ_OVERRIDE_DATA_PATH** .
   - En una instalación de IBM MQ : *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`

3. Opcional: Si ha cambiado valores de configuración en `mqwebuser.xml`, cópielos en el archivo de ejemplo.
4. Suprima el archivo `mqwebuser.xml` existente y cambie el nombre del archivo de ejemplo a `mqwebuser.xml`.

Qué hacer a continuación

Elija cómo se autentican los usuarios:

Opciones de autenticación de IBM MQ Console

- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario especifica un ID de usuario y una contraseña en la pantalla de inicio de sesión de IBM MQ Console. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. No se requiere ninguna configuración adicional para utilizar esta opción de autenticación, pero puede configurar de manera opcional el intervalo de caducidad de la señal LTPA. Si desea más información, consulte Configuración del intervalo de caducidad de señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en IBM MQ Console, sino que utiliza el certificado de cliente. Para obtener más información, consulte “Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console” en la página 535.

Opciones de autenticación de REST API

- Los usuarios se autentican utilizando la autenticación básica HTTP. En este caso, se codifican un nombre de usuario y una contraseña, pero no se cifran, y se envían a cada solicitud REST API para autenticar y autorizar al usuario para esa solicitud. Para que esta autenticación sea segura, debe utilizar una conexión segura. Es decir, debe utilizar HTTPS. Para obtener más información, consulte “Utilización de la autenticación básica HTTP con REST API” en la página 538.
- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario proporciona un ID de usuario y una contraseña al recurso REST API `login` con el método HTTP POST. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. Para obtener más información, consulte “Utilización de la autenticación basada en señal con la API REST” en la página 539. Puede configurar el intervalo de caducidad de la señal LTPA. Para obtener más información, consulte Configuración de la señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en REST API, sino que utiliza el certificado de cliente. Para obtener más información, consulte “Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console” en la página 535.

Configuración de un registro LDAP para la IBM MQ Console y la REST API

Puede configurar un registro LDAP dentro del archivo `mqwebuser.xml`. Los nombres de usuario y las contraseñas del registro LDAP se utilizan para autenticar y autorizar usuarios de la IBM MQ Console y la REST API.

Antes de empezar

- Al configurar un registro LDAP, debe asignar a cada usuario un rol. Cada rol proporciona distintos niveles de privilegio para acceder a IBM MQ Console y REST API, y determina el contexto de seguridad que se utiliza cuando se intenta una operación permitida. Tendrá que comprender estos roles antes de configurar el registro. Si desea más información sobre cada uno de los roles, consulte “Roles en IBM MQ Console y REST API” en la página 531.

  Tenga en cuenta que cualquier usuario con el rol `MQWebUser` solo puede realizar las operaciones otorgadas al ID de usuario en el gestor de colas. Por lo tanto, el ID de usuario definido en el servidor LDAP debe tener un ID de usuario idéntico en el sistema en el cual está instalado IBM MQ. Estos ID de usuario tienen que coincidir en mayúsculas y minúsculas o la correlación entre ellos puede fallar.

- Para completar esta tarea, debe ser un usuario con privilegios suficientes para editar el archivo `mqwebuser.xml`:
  - z/OS En z/OS, debe tener acceso de escritura en el archivo `mqwebuser.xml`.
  - Multi En todos los demás sistemas operativos, debe ser un usuario con privilegios.
  - Linux V 9.4.0 Si el servidor mqweb forma parte de una instalación autónoma de IBM MQ Web Server , debe tener acceso de escritura al archivo `mqwebuser.xml` en el directorio de datos IBM MQ Web Server .

Procedimiento

1. Copie el archivo XML de ejemplo `ldap_registry.xml` de una de las vías de acceso siguientes:
   - En una instalación de IBM MQ :
     - ALW En AIX, Linux, and Windows: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS En z/OS: `PathPrefix /web/mq/samp/configuration`

       donde `PathPrefix` es la vía de acceso de instalación de IBM MQ for z/OS UNIX System Services Components .
   - Linux V 9.4.0 En una instalación autónoma de IBM MQ Web Server : *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     donde *MQWEB_INSTALLATION_PATH* es el directorio en el que se ha descomprimido el archivo de instalación de IBM MQ Web Server .
2. Coloque el archivo de ejemplo en el directorio adecuado:
   - En una instalación de IBM MQ :
     - Linux AIX En AIX o Linux: `/var/mqm/web/installations/`*installationName*`/servers/mqweb`
     - Windows En Windows: *MQ_DATA_PATH*`\web\installations\`*installationName*`\servers\mqweb`, donde *MQ_DATA_PATH* es la vía de acceso de datos de IBM MQ . Esta vía de acceso es la vía de acceso de datos que se selecciona durante la instalación de IBM MQ. De forma predeterminada, esta vía de acceso es `C:\ProgramData\IBM\MQ`.
     - z/OS En z/OS: *WLP_user_directory*`/servers/mqweb`

       donde *directorio_usuario_WLP* es el directorio que se ha especificado cuando se ejecutó el script **crtmqweb** para crear la definición del servidor mqweb.
   - Linux V 9.4.0 En una instalación autónoma de IBM MQ Web Server : *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     donde *MQ_OVERRIDE_DATA_PATH* es el directorio de datos de IBM MQ Web Server al que apunta la variable de entorno **MQ_OVERRIDE_DATA_PATH** .
3. Opcional: Si ha cambiado valores de configuración en `mqwebuser.xml`, cópielos en el archivo de ejemplo.
4. Suprima el archivo `mqwebuser.xml` existente y cambie el nombre del archivo de ejemplo a `mqwebuser.xml`.
5. Edite el nuevo archivo `mqwebuser.xml` para cambiar los valores del registro LDAP en los códigos **ldapRegistry** y **idsLdapFilterProperties**.

   Si desea más información sobre cómo configurar registros LDAP, consulte Configuración de registros de usuarios LDAP en Liberty en la documentación de WebSphere Liberty.

6. Asigne roles a usuarios y grupos editando el archivo `mqwebuser.xml`:

   Existen varios roles disponibles que autorizan a usuarios y grupos a utilizar IBM MQ Console, y REST API. Cada rol otorga un nivel de acceso diferente. Para obtener más información, consulte "Roles en IBM MQ Console y REST API" en la página 531.

   - Para asignar roles y otorgar acceso a IBM MQ Console, añada los usuarios y los grupos entre las etiquetas **`security-role`** adecuadas en las etiquetas **`<enterpriseApplication id="com.ibm.mq.console">`**.
   - Para asignar roles y otorgar acceso a REST API, añada los usuarios y los grupos entre las etiquetas **`security-role`** adecuadas en las etiquetas **`<enterpriseApplication id="com.ibm.mq.rest">`**.

Qué hacer a continuación

Elija cómo se autentican los usuarios:

Opciones de autenticación de IBM MQ Console

- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario especifica un ID de usuario y una contraseña en la pantalla de inicio de sesión de IBM MQ Console. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. No se requiere ninguna configuración adicional para utilizar esta opción de autenticación, pero puede configurar de manera opcional el intervalo de caducidad de la señal LTPA. Si desea más información, consulte Configuración del intervalo de caducidad de señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en IBM MQ Console, sino que utiliza el certificado de cliente. Para obtener más información, consulte "Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console" en la página 535.

Opciones de autenticación de REST API

- Los usuarios se autentican utilizando la autenticación básica HTTP. En este caso, se codifican un nombre de usuario y una contraseña, pero no se cifran, y se envían a cada solicitud REST API para autenticar y autorizar al usuario para esa solicitud. Para que esta autenticación sea segura, debe utilizar una conexión segura. Es decir, debe utilizar HTTPS. Para obtener más información, consulte "Utilización de la autenticación básica HTTP con REST API" en la página 538.
- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario proporciona un ID de usuario y una contraseña al recurso REST API `login` con el método HTTP POST. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. Para obtener más información, consulte "Utilización de la autenticación basada en señal con la API REST" en la página 539. Puede configurar el intervalo de caducidad de la señal LTPA. Para obtener más información, consulte Configuración de la señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en REST API, sino que utiliza el certificado de cliente. Para obtener más información, consulte "Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console" en la página 535.

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 "Roles en IBM MQ Console y REST API" on page 531.

- 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 529.
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 529. 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 530. For more information about the roles, see “Roles en IBM MQ Console y REST API” on page 531.

    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

Elija cómo se autentican los usuarios:

Opciones de autenticación de IBM MQ Console

- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario especifica un ID de usuario y una contraseña en la pantalla de inicio de sesión de IBM MQ Console. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. No se requiere ninguna configuración adicional para utilizar esta opción de autenticación, pero puede configurar de manera opcional el intervalo de caducidad de la señal LTPA. Si desea más información, consulte Configuración del intervalo de caducidad de señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en IBM MQ Console, sino que utiliza el certificado de cliente. Para obtener más información, consulte “Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console” on page 535.

Opciones de autenticación de REST API

- Los usuarios se autentican utilizando la autenticación básica HTTP. En este caso, se codifican un nombre de usuario y una contraseña, pero no se cifran, y se envían a cada solicitud REST API para autenticar y autorizar al usuario para esa solicitud. Para que esta autenticación sea segura, debe

utilizar una conexión segura. Es decir, debe utilizar HTTPS. Para obtener más información, consulte “Utilización de la autenticación básica HTTP con REST API” on page 538.

- Los usuarios se autentican utilizando la autenticación de señal. En este caso, un usuario proporciona un ID de usuario y una contraseña al recurso REST API `login` con el método HTTP POST. Se genera una señal LTPA que permite al usuario permanecer conectado y autorizado durante una cantidad de tiempo definida. Para obtener más información, consulte “Utilización de la autenticación basada en señal con la API REST” on page 539. Puede configurar el intervalo de caducidad de la señal LTPA. Para obtener más información, consulte Configuración de la señal LTPA.
- Los usuarios se autentican utilizando certificados de cliente. En este caso, el usuario no utiliza un ID de usuario o una contraseña para iniciar una sesión en REST API, sino que utiliza el certificado de cliente. Para obtener más información, consulte “Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console” on page 535.

Roles en IBM MQ Console y REST API

Cuando autoriza a usuarios y grupos a utilizar la IBM MQ Console o REST API, debe asignar a los usuarios y grupos uno de los roles disponibles: **MQWebAdmin**, **MQWebAdminRO**, **MQWebUser**, **MFTWebAdmin** y **MFTWebAdminRO**. Cada rol proporciona distintos niveles de privilegio para acceder a IBM MQ Console y REST API, y determina el contexto de seguridad que se utiliza cuando se intenta una operación permitida.

Nota: Con la excepción del rol **MQWebUser** , el ID de usuario no distingue entre mayúsculas y minúsculas. Consulte “MQWebUser” en la página 531 para conocer los requisitos específicos de este rol.

MQWebAdmin
Un usuario o grupo que tiene asignado este rol puede realizar todas las operaciones administrativas y funciona bajo el contexto de seguridad del ID de usuario del sistema operativo que se ha utilizado para iniciar el servidor mqweb.

Un usuario o grupo con este rol no tiene acceso a los servicios REST siguientes:

- La REST API para MFT. Para utilizar estos servicios, el usuario o grupo también debe tener asignado el rol **MFTWebAdmin** o **MFTWebAdminRO**.
- messaging REST API. Para utilizar la messaging REST API, el usuario debe tener asignado el rol **MQWebUser**.

MQWebAdminRO
Este rol proporciona acceso de sólo lectura a la IBM MQ Console o la REST API. Un usuario o grupo al que se asigna este rol puede realizar las operaciones siguientes:

- Visualizar y consultar las operaciones en objetos de IBM MQ, como colas y canales.
- Examinar mensajes en colas.

Un usuario o grupo al que se asigna este rol opera bajo el contexto de seguridad del ID de usuario de sistema operativo que se utiliza para iniciar el servidor mqweb.

Un usuario o grupo con este rol no tiene acceso a los servicios REST siguientes:

- La REST API para MFT. Para utilizar estos servicios, el usuario o grupo también debe tener asignado el rol **MFTWebAdmin** o **MFTWebAdminRO**.
- messaging REST API. Para utilizar la messaging REST API, el usuario debe tener asignado el rol **MQWebUser**.

MQWebUser
Un usuario o grupo al que se asigne este rol podrá realizar cualquier operación permitida al ID de usuario en el gestor de colas. Por ejemplo:

- Iniciar y detener las operaciones en objetos de IBM MQ como canales.
- Definir y establecer operaciones en objetos de IBM MQ como colas y canales.
- Visualizar y consultar las operaciones en objetos de IBM MQ, como colas y canales.
- Coloque y obtenga mensajes utilizando la messaging REST API.

Un usuario o grupo al que se asigna este rol opera bajo el contexto de seguridad del principal y sólo puede realizar las operaciones que el ID de usuario está autorizado a realizar en el gestor de colas.

Por lo tanto, al usuario o al grupo que se define en el registro de usuarios mqweb debe otorgársele autorización en IBM MQ para que el usuario pueda realizar alguna operación. Mediante este rol, puede controlar con precisión qué usuarios tienen qué tipo de acceso a recursos específicos de IBM MQ cuando utilizan IBM MQ Console y REST API.

Nota:

- La longitud máxima de un ID de usuario al que se asigna este rol es de 12 caracteres.
- El ID de usuario tiene que coincidir en mayúsculas y minúsculas con del registro de usuarios de mqweb y en el sistema IBM MQ. Si las mayúsculas y minúsculas del ID de usuario son diferentes, el usuario podría ser autenticado por la IBM MQ Console y la REST API, pero no estar autorizado para utilizar recursos de IBM MQ.

MFTWebAdmin
Un usuario o grupo asignado a este rol puede realizar todas las operaciones REST de MFT y opera bajo el contexto de seguridad del ID de usuario del sistema operativo que se utiliza para iniciar el servidor `mqweb`.

Un usuario o grupo con este rol no tiene acceso a ninguno de los servicios de IBM MQ REST API. Para utilizar estos servicios, el usuario o grupo también debe tener asignado el rol **MQWebAdmin**, **MQWebAdminRO** o **MQWebUser**.

MFTWebAdminRO
Este rol proporciona acceso de sólo lectura a la REST API para MFT . Un usuario o grupo que tiene asignado este rol puede realizar operaciones de sólo lectura (solicitudes GET) como listar transferencia y listar agentes.

Un usuario o grupo al que se asigna este rol opera bajo el contexto de seguridad del ID de usuario de sistema operativo que se utiliza para iniciar el servidor mqweb.

Un usuario o grupo con este rol no tiene acceso a ninguno de los servicios de IBM MQ REST API. Para utilizar estos servicios, el usuario o grupo también debe tener asignado el rol **MQWebAdmin**, **MQWebAdminRO** o **MQWebUser**.

Para obtener más información sobre cómo configurar usuarios y grupos para utilizar estos roles, consulte “Configuración de usuarios y roles” en la página 520.

Solapamiento de roles

A un usuario o grupo se le puede asignar más de un rol. Cuando un usuario realiza una operación en esta situación, se utiliza el rol de mayor privilegio que es aplicable a la operación. Por ejemplo, si un usuario con los roles **MQWebAdminRO** y **MQWebUser** realiza una operación de cola de consulta, se utiliza el rol **MQWebAdminRO** y se intenta la operación bajo el contexto del ID de usuario de sistema que ha iniciado el servidor web. Si dicho mismo usuario realiza una operación de definición, se utiliza el rol **MQWebUser** y se intenta la operación bajo el contexto del principal.

ALW Cambio del certificado presentado por IBM MQ Console en el navegador

Puede configurar IBM MQ Console para presentar un certificado firmado por CA para fines de autenticación. Si configura IBM MQ Console para presentar un certificado firmado por CA, el navegador ya no presenta el aviso de certificado autofirmado cuando se accede a IBM MQ Console .

Acerca de esta tarea

La seguridad para IBM MQ Console la proporciona el servidor mqweb que ejecuta IBM MQ Console. Para cambiar el certificado que el servidor mqweb presenta al navegador, primero añada el nuevo certificado al almacén de claves del servidor mqweb. A continuación, edite la configuración de seguridad en el archivo `mqwebuser.xml` para especificar el certificado que presenta el servidor.

El procedimiento presupone lo siguiente:

- Que es un usuario privilegiado.
- Está utilizando un sistema AIX, Linuxo Windows .
- Que el archivo `mqwebuser.xml` se basa en los archivos XML de ejemplo `basic_registry.xml`, `local_os_registry.xml`o `ldap_registry.xml` .

Procedimiento

1. Opcional: Cambie la contraseña predeterminada del almacén de claves del servidor mqweb `key.jks` utilizando el mandato **runmqktool** :

```
runmqktool -storepasswd -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass oldPassword
          -new newPassword
```

   oldPassword
   Especifica la contraseña de `key.jks` existente. La contraseña predeterminada es `password`.

   newPassword
   Especifica una nueva contraseña de `key.jks` .

2. Cree un par de claves y una solicitud de certificado para enviar a la entidad emisora de certificados:

   a) Cree el par de claves utilizando el mandato **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
```

   password
   Especifica la contraseña del almacén de claves de `key.jks` .

   label
   Especifica la etiqueta de certificado. Por ejemplo, `MQWebConsole`.

   nombre_distinguido
   Especifica el nombre distinguido X.500 para el certificado. Escriba el nombre distinguido entre comillas dobles.

   Por ejemplo: `"cn=MQWebConsole,o=myOrg,c=UK"`

   signature_algorithm
   Especifica el algoritmo que se debe utilizar para firmar el certificado. Para obtener más información, consulte Algoritmos de firma

   b) Cree la solicitud de certificado utilizando el mandato **runmqktool** :

```
runmqktool -certreq -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password -alias label
           -file filename
```

   password
   Especifica la contraseña del almacén de claves de `key.jks` .

   label
   Especifica la etiqueta de certificado del subpaso “2.a” en la página 533.

   nombrearchivo
   Especifica el nombre de archivo completo para la solicitud de certificado.

3. Envíe el archivo de solicitud de certificado a una entidad emisora de certificados (CA).
4. Cuando tenga el certificado de la CA, importe el certificado y cualquier otro certificado de la cadena de certificados, empezando por el certificado de la CA raíz, en el almacén de claves de `keys.jks` utilizando el mandato **runmqktool** :

```
runmqktool -importcert -keystore MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/key.jks -storepass password
           -alias label -file filename
```

password
Especifica la contraseña del almacén de claves de `key.jks` .

label
Especifica la etiqueta de certificado del subpaso “2.a” en la página 533.

nombrearchivo
Especifica el nombre de archivo completo del certificado que se va a importar.

5. Configure el servidor mqweb para presentar el certificado de CA:
   a) Abra el archivo `mqwebuser.xml`.

      El archivo `mqwebuser.xml` se puede encontrar en la siguiente vía de acceso: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`
   b) Desactive la configuración de seguridad predeterminada comentando la línea siguiente:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

      Si ha configurado el servidor mqweb para utilizar la autenticación de certificado de cliente, esta línea del archivo xml ya está comentada.
   c) Elimine el comentario de la sección del archivo `mqwebuser.xml` que habilita la configuración de certificados personalizados. La sección contiene el texto siguiente:

```
<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"/>
```

      Si ha configurado el servidor mqweb para utilizar la autenticación de certificado de cliente, esta sección del archivo xml ya no está comentada.
   d) Opcional: Si ha cambiado la contraseña para el almacén de claves de `key.jks` en el paso “1” en la página 533, cambie el valor de **password** en las etiquetas `defaultKeyStore` a una versión codificada de la contraseña que ha establecido:
      i) En el directorio *MQ_INSTALLATION_PATH*`/web/bin` , especifique el mandato siguiente:

```
securityUtility encode password
```

      ii) Coloque la salida de este mandato en el campo **password** de `defaultKeyStore`.
   e) Si no está utilizando la autenticación de certificado de cliente, comente la línea siguiente:

```
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
```

   f) Cambie el valor de **serverKeyAlias** de `default` al valor de la etiqueta de certificado de CA.
6. Detenga el servidor mqweb utilizando el mandato **endmqweb** .
7. Inicie el servidor mqweb utilizando el mandato **strmqweb** .

Resultados

Cuando se inicie el servidor web, vaya a IBM MQ Console y renueve. Se utiliza el certificado de CA y se le lleva directamente a la página de inicio de sesión.

ALW Configuración de la autenticación de certificados de cliente con REST API y IBM MQ Console

Puede correlacionar certificados de cliente con principales para autenticar usuarios de IBM MQ Console y REST API.

Antes de empezar

- Configure los usuarios, los grupos y los roles a los que se les va a autorizar el uso de IBM MQ Console y REST API. Para obtener más información, consulte “Configuración de usuarios y roles” en la página 520.
- Cuando utiliza REST API, puede consultar las credenciales del usuario actual utilizando el método HTTP GET en el recurso `login`, proporcionando el certificado de cliente para autenticar la solicitud. Esta solicitud devuelve información sobre el nombre de usuario y los roles que se han asignado al usuario. Para obtener más información, consulte GET `/login`.
- Cuando los certificados de cliente se correlacionan con principales para autenticar usuarios, se utiliza el nombre distinguido del certificado de cliente para compararlo con los usuarios del registro de usuarios configurado:
  - Para un registro básico, se compara el nombre común (CN) con el usuario. Por ejemplo, `CN=Fred, O=IBM, C=GB` se compara con un nombre de usuario de `Fred`.
  - Para un registro LDAP, el nombre distinguido completo se compara con LDAP de forma predeterminada. Puede configurar filtros y correlaciones para personalizar la coincidencia. Si desea más información, consulte Liberty:Modalidad de correlación de certificado LDAP en la documentación de WebSphere Liberty.

Acerca de esta tarea

Cuando un usuario se autentica utilizando un certificado de cliente, se utiliza el certificado en lugar de un nombre de usuario y una contraseña. Para REST API, el certificado de cliente se proporciona con cada solicitud REST para autenticar el usuario. Para IBM MQ Console, cuando un usuario inicia una sesión con un certificado, no puede cerrarla.

ALW En sistemas AIX, Linuxo Windows , el procedimiento presupone la información siguiente:

- Que el archivo `mqwebuser.xml` se basa en los archivos XML de ejemplo `basic_registry.xml`, `local_os_registry.xmlo ldap_registry.xml` .
- Que es un usuario privilegiado.

z/OS Para configurar la autenticación de certificados de cliente con un conjunto de claves RACF en sistemas z/OS , siga el procedimiento de “Configuring TLS for the REST API and IBM MQ Console on z/OS” en la página 547.

Nota: En el siguiente procedimiento se describen los pasos necesarios para utilizar certificados de cliente con IBM MQ Console y REST API. Para la comodidad del desarrollador, los pasos detallan cómo crear y utilizar certificados autofirmados. No obstante, para la producción, utilice certificados obtenidos de una entidad emisora de certificados.

Procedimiento

1. Cree un certificado utilizando el mandato **`runmqktool`** :

```
runmqktool -genkeypair -keystore filename -storepass password -storetype PKCS12
           -alias label -dname distinguished_name
           -sigalg signature_algorithm
```

nombrearchivo
Especifica el nombre del almacén de claves, por ejemplo `user.p12`. Si el almacén de claves no existe, se crea cuando se ejecuta el mandato.

password
Especifica la contraseña del almacén de claves.

label
Especifica la etiqueta de certificado. Por ejemplo, `user1`.

nombre_distinguido
Especifica el nombre distinguido X.500 para el certificado. Escriba el nombre distinguido entre comillas dobles.

Si está utilizando un registro de usuarios básico, especifique el nombre de un usuario del registro de usuarios en la parte Nombre común (CN) del nombre distinguido. Por ejemplo, para un usuario `mqadmin`, utilice el nombre distinguido `"CN=mqadmin"`.

Si está utilizando un registro de sistema operativo local, especifique el nombre de un ID de usuario de sistema operativo local en la parte Nombre común (CN) del nombre distinguido. Por ejemplo, para un usuario `mqadmin`, utilice el nombre distinguido `"CN=mqadmin"`.

Si está utilizando un registro de usuarios LDAP, especifique un nombre distinguido que coincida con el nombre distinguido en el registro LDAP.

signature_algorithm
Especifica el algoritmo que se debe utilizar para firmar el certificado. Para obtener más información, consulte Algoritmos de firma

2. Opcional: Obtenga un certificado de una entidad emisora de certificados (CA). De forma alternativa, para utilizar un certificado autofirmado, continúe en el paso “3” en la página 536.
   a) Para obtener un certificado de una entidad emisora de certificados, cree una solicitud de certificado utilizando el mandato **runmqktool** :

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

   nombrearchivo
   Especifica el nombre del almacén de claves del paso “1” en la página 535.

   password
   Especifica la contraseña del almacén de claves.

   label
   Especifica la etiqueta de certificado del paso “1” en la página 535.

   nombrearchivo
   Especifica el nombre de archivo completo para la solicitud de certificado.

   b) Envíe el archivo de solicitud de certificado a una entidad emisora de certificados (CA).

   c) Cuando tenga el certificado de la CA, importe el certificado en el almacén de claves utilizando el mandato **runmqktool** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

   nombrearchivo
   Especifica el nombre del almacén de claves del paso “1” en la página 535.

   password
   Especifica la contraseña del almacén de claves.

   label
   Especifica la etiqueta de certificado del paso “1” en la página 535.

   nombrearchivo
   Especifica el nombre de archivo completo del certificado de CA.

3. Extraiga la parte pública del certificado utilizando el mandato **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass password
           -alias label -file filename -rfc
```

nombrearchivo
Especifica el nombre del almacén de claves del paso "1" en la página 535.

password
Especifica la contraseña del almacén de claves.

label
Especifica la etiqueta de certificado del paso "1" en la página 535.

nombrearchivo
Especifica el nombre de archivo completo para el certificado extraído.

4. Importe la parte pública del certificado en el almacén de claves de confianza del servidor mqweb como un certificado de firmante para que el servidor pueda validar el certificado de cliente utilizando el mandato **runmqktool** :

```
runmqktool -importcert -keystore  MQ_DATA_DIRECTORY/web/installations/installationName/
servers/mqweb/resources/security/trust.jks -storepass password
           -alias label -file filename
```

password
Especifica la contraseña del almacén de claves de `trust.jks` . Puede especificar una contraseña para un almacén de claves `trust.jks` existente o una contraseña nueva para un almacén de claves `trust.jks` nuevo.

label
Especifica la etiqueta de certificado del paso "1" en la página 535.

nombrearchivo
Especifica el nombre de archivo completo del certificado extraído.

5. Configure el servidor mqweb para que utilice la autenticación de certificado de cliente:
   a) Abra el archivo `mqwebuser.xml`.

      El archivo `mqwebuser.xml` se puede encontrar en la siguiente vía de acceso: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

   b) Desactive la configuración de seguridad predeterminada comentando la línea siguiente:

```
<sslDefault sslRef="mqDefaultSSLConfig"/>
```

      Si ha configurado el servidor mqweb para presentar un certificado de CA al navegador, esta línea ya está comentada.

   c) Elimine el comentario de la sección del archivo `mqwebuser.xml` que permite la autenticación de certificados de cliente. La sección contiene el texto siguiente:

```
<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"/>
```

      Si ha configurado el servidor mqweb para presentar un certificado de CA al navegador, esta sección ya no está comentada. Sin embargo, es posible que tenga que descomentar la línea **defaultTrustStore** .

   d) Cambie el valor de **password** en `defaultTrustStore` para que coincida con la contraseña del almacén de claves `trust.jks`:
      i) En el directorio *MQ_INSTALLATION_PATH*/web/bin , especifique el mandato siguiente:

```
securityUtility encode password
```

      ii) Coloque la salida de este mandato en el campo **password** de `defaultTrustStore`.

6. Detenga el servidor mqweb utilizando el mandato **endmqweb** .

7. Inicie el servidor mqweb utilizando el mandato **strmqweb** .
8. Utilice el certificado de cliente para la autenticación:
   - Para utilizar el certificado de cliente con IBM MQ Console, instale el certificado de cliente en el navegador web que se utiliza para acceder a IBM MQ Console.
   - Para utilizar el certificado de cliente con REST API, proporcione el certificado de cliente con cada solicitud REST. Cuando utiliza los métodos HTTP POST, PATCH o DELETE, debe proporcionar una autenticación adicional con el certificado de cliente para evitar ataques de falsificación de solicitudes entre sitios. Es decir, la autenticación adicional se utiliza para confirmar que las credenciales para autenticar la solicitud las utiliza el propietario de las credenciales.

     Esta autenticación adicional la proporciona la cabecera HTTP `ibm-mq-rest-csrf-token`. Establezca el valor de la cabecera `ibm-mq-csrf-token` en cualquier valor, incluido en blanco y, a continuación, envíe la solicitud.

Ejemplo

Importante: En el ejemplo, no todas las implementaciones de cURL dan soporte a certificados autofirmados, por lo que debe utilizar una implementación de cURL que sí lo haga.

El siguiente ejemplo de cURL muestra cómo crear una cola nueva `Q1`, en un gestor de colas `QM1`, con autenticación de certificado de cliente. La configuración exacta de este mandato cURL depende de las bibliotecas con las que se ha creado cURL . El ejemplo se basa en un sistema Windows con cURL creado en OpenSSL.

- Utilice el método HTTP POST con el recurso de cola y autentíquese con el certificado de cliente, incluyendo el contenido de la cabecera HTTP `ibm-mq-rest-csrf-token` con un valor arbitrario. Este valor puede ser cualquier valor, incluido el espacio en blanco. El distintivo `--cert-type` especifica que el certificado es un certificado PKCS#12. El distintivo `--cert` especifica la ubicación del certificado, seguido de dos puntos y, a continuación, la contraseña del 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\"}"
```

Utilización de la autenticación básica HTTP con REST API

Los usuarios de REST API pueden autenticarse proporcionando su ID de usuario y su contraseña en una cabecera HTTP. Para utilizar este método de autenticación con métodos HTTP como, por ejemplo, POST, PATCH y DELETE, la cabecera HTTP `ibm-mq-rest-csrf-token` también se debe proporcionar, así como un ID de usuario y una contraseña.

Antes de empezar

- Configure los usuarios, grupos y roles a los que se les va a autorizar a utilizar REST API. Para obtener más información, consulte “Configuración de usuarios y roles” en la página 520.
- Asegúrese de que la autenticación básica HTTP esté habilitada. Compruebe que el siguiente XML esté presente y no comentado en el archivo `mqwebuser.xml`. El XML debe estar dentro de las etiquetas `<featureManager>`:

```
<feature>basicAuthenticationMQ-1.0</feature>
```

z/OS En z/OS, debe ser un usuario que tenga acceso de escritura a `mqwebuser.xml` para editar este archivo.

Multi En todos los demás sistemas operativos, debe ser un usuario privilegiado para editar el archivo `mqwebuser.xml`.

- Asegúrese de que esté utilizando una conexión segura cuando envíe solicitudes REST. Como la combinación de nombre de usuario y contraseña está codificada, pero no cifrada, debe utilizar una conexión segura (HTTPS) cuando se utiliza la autenticación básica HTTP con REST API.
- Para consultar las credenciales del usuario actual, utilice el método HTTP GET en el recurso `login` y proporcione la información de autenticación básica para autenticar la solicitud. Esta solicitud devuelve información sobre el nombre de usuario, y los roles a los que está asignado el usuario. Para obtener más información, consulte GET `/login`.

Procedimiento

1. Concatene el nombre de usuario con dos puntos y la contraseña. Tenga en cuenta que el nombre de usuario distingue entre mayúsculas y minúsculas.

   Por ejemplo, un nombre de usuario admin y una contraseña admin se convierten en la siguiente serie:

   ```
   admin:admin
   ```

2. Codifique esta serie de nombre de usuario y contraseña en codificación base64.
3. Incluya este nombre de usuario y contraseña codificados en una cabecera HTTP `Authorization: Basic`.

   Por ejemplo, con un nombre de usuario codificado admin y una contraseña admin, se crea la siguiente cabecera:

   ```
   Authorization: Basic YWRtaW46YWRtaW4=
   ```

4. Cuando utiliza los métodos HTTP POST, PATCH o DELETE, debe proporcionar una autenticación adicional, así como un nombre de usuario y una contraseña.

   Esta autenticación adicional la proporciona la cabecera HTTP `ibm-mq-rest-csrf-token`. La cabecera HTTP `ibm-mq-rest-csrf-token` debe estar presente en la solicitud, pero su valor puede ser cualquier valor, incluyendo espacios en blanco.
5. Envíe la solicitud REST a IBM MQ con las cabeceras correspondientes.

Ejemplo

El siguiente ejemplo muestra cómo crear una nueva cola Q1 en el gestor de colas QM1, con la autenticación básica, en sistemas Windows. El ejemplo utiliza el cURL:

- Utilice el método HTTP POST con el recurso de cola y autentíquese con la autenticación básica, incluyendo la cabecera HTTP `ibm-mq-rest-csrf-token` con un valor arbitrario. Este valor puede ser cualquier valor, incluyendo los espacios en blanco:

```
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\"}"
```

Utilización de la autenticación basada en señal con la API REST

Los usuarios de REST API pueden autenticarse proporcionando un ID de usuario y una contraseña al recurso REST API `login` con el método HTTP POST. Se genera una señal LTPA que permite al usuario autenticar solicitudes en el futuro. Esta señal LTPA tiene el prefijo `LtpaToken2`. El usuario puede finalizar la sesión utilizando el método HTTP DELETE y puede consultar la información de inicio de sesión del usuario actual con el método HTTP GET.

Antes de empezar

- Configure los usuarios, grupos y roles a los que se les va a autorizar a utilizar REST API. Para obtener más información, consulte “Configuración de usuarios y roles” en la página 520.

- De forma predeterminada, el nombre de la cookie que incluye la señal LTPA empieza con `LtpaToken2` e incluye un sufijo que puede cambiar cuando se reinicia el servidor mqweb. Este nombre de cookie aleatorizado permite que se pueda ejecutar más de un servidor mqweb en el mismo sistema. Sin embargo, si desea que el nombre de la cookie siga siendo un valor coherente, puede especificar el nombre que tiene la cookie utilizando el mandato **setmqweb**. Para obtener más información, consulte Configuración de la señal LTPA.
- De forma predeterminada, la cookie de la señal LTPA caduca después de 120 minutos. Puede configurar la hora de caducidad de la cookie de la señal LTPA utilizando el mandato **setmqweb**. Para obtener más información, consulte Configuración de la señal LTPA.
- Asegúrese de que esté utilizando una conexión segura cuando envíe solicitudes REST. Cuando utiliza el método HTTP POST en el recurso `login`, la combinación de nombre de usuario y contraseña que se envía con la solicitud no están cifrados. Por lo tanto, debe utilizar una conexión segura (HTTPS) cuando utiliza la autenticación basada en señal con REST API. De forma predeterminada, no puede utilizar HTTP con la autenticacion de señal LTPA. Puede habilitar la señal LTPA para que sea utilizada por conexiones HTTP no seguras estableciendo **secureLTPA** en `False`. Para obtener más información, consulte Configuración de la señal LTPA.
- Puede consultar las credenciales del usuario actual utilizando el método HTTP GET en el recurso `login`, proporcionando la señal LTPA para autenticar la solicitud. Esta solicitud devuelve información sobre el nombre de usuario, y los roles a los que está asignado el usuario. Para obtener más información, consulte GET `/login`.

Procedimiento

1. Inicie una sesión de un usuario:
   a) Utilice el método HTTP POST en el recurso `login`:

   ```
   https://host:port/ibmmq/rest/v1/login
   ```

   Incluya el nombre de usuario y la contraseña en el cuerpo de la solicitud JSON, con el formato siguiente:

   ```
   {
       "username" : name,
       "password" : password
   }
   ```

   b) Almacene la señal LTPA que se ha devuelto de la solicitud en el almacén de cookies local. De forma predeterminada, esta señal LTPA tiene un prefijo de `LtpaToken2`.
2. Autentique las solicitudes REST con la señal LTPA almacenada como una cookie con cada solicitud.

   Para las solicitudes que utilizan los métodos HTTP PUT, PATCH o DELETE, incluya una cabecera `ibm-mq-rest-csrf-token`. El valor de esta cabecera puede ser cualquier elemento, incluso estar en blanco.
3. Cierre la sesión de un usuario:
   a) Utilice el método HTTP DELETE en el recurso `login`:

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

   Debe proporcionar la señal LTPA como una cookie para autenticar la solicitud e incluir una cabecera `ibm-mq-rest-csrf-token`. El valor de esta cabecera puede ser cualquier elemento, incluso estar en blanco.

   b) Procese la instrucción para suprimir la señal LTPA del almacén de cookies local.

   Nota: Si la instrucción no se procesa y la señal LTPA permanece en el almacén de cookies local, la señal LTPA puede utilizarse para autenticar solicitudes REST en el futuro. Es decir, cuando el usuario intenta autenticarse con la señal LTPA una vez finalizada la sesión, se crea una nueva sesión que utiliza la señal existente.

Ejemplo

El siguiente ejemplo de cURL muestra cómo crear una nueva cola Q1 en el gestor de colas QM1, con la autenticación basada en señal, en sistemas Windows:

- Inicie sesión y añada la señal LTPA con el prefijo `LtpaToken2` al almacén de cookies local. La información de nombre de usuario y contraseña se incluye en el cuerpo JSON. El distintivo `-c` especifica la ubicación del archivo donde se almacena la señal:

```
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
```

- Cree una cola. Utilice el método HTTP POST con el recurso de cola y autentíquese con la señal LTPA. La señal LTPA con el prefijo `LtpaToken2` se recupera del archivo `cookiejar.txt` utilizando el código `-b`. La presencia de la cabecera HTTP `ibm-mq-rest-csrf-token` proporcionan protección CSRF:

```
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\"}"
```

- Cierre la sesión y suprima la señal LTPA del almacén de cookies local. La señal LTPA se recupera del archivo `cookiejar.txt` utilizando el código `-b`. La presencia de la cabecera HTTP `ibm-mq-rest-csrf-token` proporciona protección CSRF. La ubicación del archivo `cookiejar.txt` se especifica mediante el distintivo `-c` para que la señal LTPA se suprima del archivo:

```
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
```

Referencia relacionada

POST/login

GET/login

Suprimir /login

Inclusión de la IBM MQ Console en un cuadro de información

Se puede utilizar el elemento HTML <iframe> para incluir una página web en otra utilizando un marco flotante (IFrame). Por razones de seguridad, la IBM MQ Console no puede estar incluida en un IFrame de forma predeterminada. Sin embargo, puede habilitar un IFrame utilizando la propiedad de configuración **mqConsoleFrameAncestors** en el servidor mqweb.

Acerca de esta tarea

El servidor mqweb mantiene una lista de elementos permitidos de orígenes de páginas web que pueden incluir IBM MQ Console utilizando un IFrame. Un origen es una combinación de un esquema, dominio y puerto de URL, por ejemplo, `https://example.com:1234`.

Puede utilizar la propiedad de configuración **mqConsoleFrameAncestors** en el servidor mqweb para especificar las entradas de la lista.

De forma predeterminada, **mqConsoleFrameAncestors** está en blanco, lo que significa que la IBM MQ Console no se puede incorporar en un IFrame.

Procedimiento

Especifique una lista de orígenes de páginas web, que puede incluir la IBM MQ Console en un IFrame, especificando el mandato siguiente:

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

donde *allowedOrigins* es una lista separada por comas de los orígenes. Cada origen debe estar formado por:

- Un nombre de host o una dirección IP
- Un esquema de URL opcional
- Un número de puerto opcional

Tenga en cuenta que el nombre de host puede empezar con el carácter comodín (*) y que el número de puerto también puede utilizar el carácter comodín (*).

Los orígenes de ejemplo son:

```
https://example.com:1234
```

que permite que cualquier página web servida desde `https://example.com:1234` incluya IBM MQ Console en un IFrame.

```
https://*.example.com:*
```

que permite que cualquier página web HTTPS con un nombre de host que termine en `example.com`, y que utilice cualquier puerto, incluya IBM MQ Console en un IFrame.

Ejemplo

El ejemplo siguiente permite que la inclusión de la IBM MQ Console en un IFrame desde páginas web proporcionadas desde `https://site2.example.com:1234` o `https://site2.example.com:1235`:

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

Configuración de CORS para REST API

De forma predeterminada, un navegador web no permite que scripts como, por ejemplo, JavaScript, invoquen REST API cuando no provienen del mismo origen que REST API. Es decir, las solicitudes entre orígenes no están habilitadas. Puede configurar Cross Origin Resource Sharing (CORS) para permitir las solicitudes entre orígenes a partir de los orígenes especificados.

Acerca de esta tarea

Puede acceder a REST API mediante un navegador web, por ejemplo, mediante un script. Como estas solicitudes son de un origen diferente a REST API, el navegador web rechaza la solicitud porque es una solicitud entre orígenes. El origen es diferente si el dominio, el puerto o el esquema no son los mismos.

Por ejemplo, si tiene un script que se aloja en `http://localhost:1999/`, realiza una solicitud entre orígenes si emite un HTTP GET en un sitio web que está alojado en `https://localhost:9443/`. Esta solicitud es una solicitud entre orígenes porque los números de puerto y el esquema (HTTP) son diferentes.

Para habilitar solicitudes entre orígenes, configure CORS y especifique los orígenes que pueden acceder REST API.

Para obtener más información sobre CORS, consulte https://www.w3.org/TR/cors/ y https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

Procedimiento

1. Consulte la configuración actual especificando el mandato siguiente:

   ```
   dspmqweb properties -a
   ```

La entrada `mqRestCorsAllowedOrigins` especifica los orígenes permitidos. La entrada `mqRestCorsMaxAgeInSeconds` especifica el tiempo, en segundos, durante el cual el navegador web puede almacenar en memoria caché los resultados de cualquier comprobación previa al lanzamiento de CORS.

2. Especifique los orígenes que pueden acceder a REST API mediante el mandato siguiente:

   ```
   setmqweb properties -k mqRestCorsAllowedOrigins -v allowedOrigins
   ```

   donde *allowedOrigins* especifica el origen desde el que se desea permitir solicitudes de orígenes. Puede utilizar un asterisco encerrado entre comillas dobles ("*") para permitir todas las peticiones entre orígenes. Puede especificar más de un origen en una lista separada por comas, encerrados entre comillas dobles. Para no permitir solicitudes entre orígenes, especifique comillas dobles vacías como valor de *allowedOrigins*.
3. Especifique el tiempo, en segundos, que desea permitir que un navegador web almacene en caché el resultado de las comprobaciones preparatorias CORS ejecutando el mandato siguiente:

   ```
   setmqweb properties -k mqRestCorsMaxAgeInSeconds -v time
   ```

Ejemplo

El siguiente ejemplo muestra las solicitudes entre orígenes habilitadas para `http://localhost:9883`, `https://localhost:1999` y `https://localhost:9663`. La antigüedad máxima de los resultados en la memoria caché de cualquier comprobación previa de CORS se establece en 90 segundos:

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://
localhost:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

Configurando la validación de la cabecera de host para la IBM MQ Console y la REST API

Puede configurar el servidor mqweb para restringir el acceso a IBM MQ Console y REST API de tal forma que sólo se procesen las solicitudes que se envían con una cabecera de host que coincida con una lista de elementos permitidos especificada. Se devuelve un error si se utiliza un valor de cabecera de host que no está en la lista de elementos permitidos.

Acerca de esta tarea

El servidor mqweb utiliza hosts virtuales para definir la lista de elementos permitidos de cabeceras de host aceptables. Para obtener más información sobre los hosts virtuales, consulte la documentación de WebSphere Liberty : https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html

Para completar esta tarea, debe ser un usuario con privilegios suficientes para editar el archivo `mqwebuser.xml`:

- z/OS En z/OS, debe tener acceso de escritura en el archivo `mqwebuser.xml`.
- Multi En todos los demás sistemas operativos, debe ser un usuario con privilegios.
- Linux V 9.4.0 Si el servidor mqweb forma parte de una instalación autónoma de IBM MQ Web Server , debe tener acceso de escritura al archivo `mqwebuser.xml` en el directorio de datos IBM MQ Web Server .

Procedimiento

1. Abra el archivo `mqwebuser.xml`. Este archivo se encuentra en una de las ubicaciones siguientes:
   - En una instalación de IBM MQ :

– Linux AIX En AIX o Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`

– Windows En Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb`, donde *MQ_DATA_PATH* es la vía de acceso de datos de IBM MQ . Esta vía de acceso es la vía de acceso de datos que se selecciona durante la instalación de IBM MQ. De forma predeterminada, esta vía de acceso es `C:\ProgramData\IBM\MQ`.

– z/OS En z/OS: *WLP_user_directory*`/servers/mqweb`

Donde *directorio_usuario_WLP* es el directorio que se ha especificado al ejecutar el mandato **crtmqweb** para crear la definición del servidor mqweb.

- Linux V 9.4.0 En una instalación autónoma de IBM MQ Web Server : *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

  donde *MQ_OVERRIDE_DATA_PATH* es el directorio de datos de IBM MQ Web Server al que apunta la variable de entorno **MQ_OVERRIDE_DATA_PATH** .

2. Añada o descomente el código siguiente en el archivo `mqwebuser.xml`:

```
<virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
        <hostAlias>localhost:9080</hostAlias>
</virtualHost>
```

3. Edite el campo **<hostAlias>** , insertando la combinación de nombre de host y puerto que desea permitir.

   Esta combinación puede ser el nombre de host y el nombre de puerto que ha utilizado en la configuración del servidor mqweb. Por ejemplo, si utiliza la configuración predeterminada de `localhost:9443`, es posible que desee utilizar `localhost:9443` en el campo **<hostAlias>** .

   Si es necesario, puede añadir varios campos **<hostAlias>** en las etiquetas **<virtualHost>** para permitir más combinaciones de nombre de host y puerto. Por ejemplo, para permitir las cabeceras de host que utilizan un puerto HTTP, así como las cabeceras de host que utilizan el puerto HTTPS.

Auditoría

Los registros de auditoría de las operaciones que se realizan en IBM MQ Console y REST API se pueden generar habilitando el mandato del gestor de colas y los sucesos de configuración, y en AIX, Linux, and Windows se registran cambios de estado significativos en los archivos de registro del servidor mqweb.

Cambios de estado significativos

ALW

En AIX, Linux, and Windows, IBM MQ Console registra los cambios de estado significativos como mensajes en los registros del servidor mqweb. Cada mensaje indica el nombre de principal autenticado que ha solicitado la operación.

Los cambios de estado significativos, por ejemplo cuando se crean, inician, finalizan o suprimen gestores de colas, se registran en los archivos `messages.log` y `console.log` del servidor mqweb a nivel de registro [AUDIT]. Cada entrada de registro indica el nombre de principal autenticado que ha solicitado la operación.

Los archivos `messages.log` y `console.log` se pueden encontrar en la ubicación siguiente:

- En una instalación de IBM MQ :

  – Linux AIX En AIX o Linux: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb/logs`

  – Windows En Windows: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb\logs`, donde

MQ_DATA_PATH es la vía de acceso de datos de IBM MQ . Esta vía de acceso es la vía de acceso de datos que se selecciona durante la instalación de IBM MQ. De forma predeterminada, esta vía de acceso es `C:\ProgramData\IBM\MQ`.

- Linux V 9.4.0 En una instalación autónoma de IBM MQ Web Server : *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb/logs`

  donde *MQ_OVERRIDE_DATA_PATH* es el directorio de datos de IBM MQ Web Server al que apunta la variable de entorno **MQ_OVERRIDE_DATA_PATH** .

Para obtener más información sobre cómo configurar los niveles de registro del servidor mqweb, consulte Configuración del registro.

Sucesos de mandato y de configuración

Opcionalmente, puede habilitar los sucesos de mandato y de configuración en el gestor de colas para proporcionar información sobre la mayor parte de la actividad de IBM MQ Console y REST API. Por ejemplo, la creación de canales y la consulta de colas generan sucesos de mandato y de configuración. Para obtener más información sobre cómo habilitar sucesos de mandato y de configuración, consulte Control de sucesos de configuración, mandato y registrador.

Para estos mensajes de suceso de mandato y configuración, el campo **MQIACF_EVENT_ORIGIN** se establece en `MQEVO_REST` y el campo **MQCACF_EVENT_APPL_IDENTITY** notifica los primeros 32 caracteres del nombre de principal autenticado. Si un usuario tiene el rol `MQWebAdmin` o `MQWebAdminRO` , el campo **MQCACF_EVENT_USER_ID** notifica el ID de usuario del servidor mqweb, no el nombre de usuario del principal que ha emitido el mandato. Sin embargo, si el usuario tiene el rol `MQWebUser` , **MQCACF_EVENT_USER_ID** notifica el nombre de usuario del principal que ha emitido el mandato.

Conceptos relacionados

“Auditoría” en la página 486
Puede comprobar las intrusiones de seguridad, o intentos de intrusión, mediante mensajes de sucesos. También puede comprobar la seguridad del sistema utilizando IBM MQ Explorer.

z/OS Consideraciones de seguridad para el iniciador de canal de IBM MQ Console y REST API en z/OS

IBM MQ Console y REST API tienen características de seguridad que controlar si un usuario puede emitir, visualizar o modificar mandatos. Los mandatos se pasan al gestor de colas y, a continuación, se utiliza la seguridad del gestor de colas para controlar si el usuario está autorizado para emitir el mandato para dicho gestor de colas específico.

Procedimiento

1. Asegúrese de que el ID de usuario de la tarea iniciada del servidor mqweb tiene las autorizaciones apropiadas para emitir determinados mandatos PCF y acceder a determinadas colas. Para obtener más información, consulte “Authority required by the mqweb server started task user ID” en la página 546.
2. Asegúrese de que todos los usuarios a los que se les ha otorgado el rol `MQWebUser` tienen las autorizaciones apropiadas.

   Los usuarios de IBM MQ Console y REST API asignados al rol `MQWebUser` funcionan bajo el contexto de seguridad del principal. Estos ID de usuario solo pueden realizar operaciones para las que se les ha otorgado autorización para realizar en el gestor de colas y deben tener acceso a las mismas colas de sistema que el espacio de direcciones de servidor mqweb.

   Al ID de usuario de la tarea iniciada del servidor mqweb se les debe haber otorgado acceso de usuario alternativo a todos los usuarios asignados al rol `MQWebUser`.

Si desea más información sobre cómo otorgar las autorizaciones apropiadas para los usuarios con el rol `MQWebUser`, consulte "Acceso a recursos de IBM MQ necesarios para utilizar la IBM MQ Console o la REST API" en la página 546.

3. Opcional: Configure TLS para la IBM MQ Console y la REST API. Para obtener más información, consulte "Configuring TLS for the REST API and IBM MQ Console on z/OS" en la página 547.

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 235, "System queue security" on page 214, and "Profiles for context security" on page 224.
- 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 205.

z/OS Acceso a recursos de IBM MQ necesarios para utilizar la IBM MQ Console o la REST API

Las operaciones realizadas en IBM MQ Console o REST API por un usuario con el rol `MQWebUser` tienen lugar en el contexto de seguridad del usuario.

Acerca de esta tarea

Consulte "Roles en IBM MQ Console y REST API" en la página 531 para obtener más información sobre los roles en IBM MQ Console y REST API.

Utilice el procedimiento siguiente para otorgar a un usuario con el rol `MQWebUser` acceso a los recursos del gestor de colas necesarios para utilizar IBM MQ Console o REST API.

Procedimiento

1. Otorgue al ID de usuario `mqweb server started task` acceso de usuario alternativo a cada ID de usuario en el rol `MQWebUser`.

   Realice esta tarea en cada uno de los gestores de colas que administrarán los usuarios mediante IBM MQ Console o REST API.

   Puede utilizar los siguientes mandatos RACF de ejemplo para otorgar al ID de usuario de `mqweb server started task` acceso de usuario alternativo a un usuario con el rol `MQWebUser` :

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   donde:

   hlq
   Es el prefijo de perfil, que puede ser el nombre del gestor de colas o el nombre del grupo de compartición de colas

   userId
   Es el usuario con el rol `MQWebUser`

   mqwebUserId
   Es el ID de usuario de `mqweb server started task`

   Nota: Si utiliza seguridad que combina mayúsculas y minúsculas, utilice la clase MXADMIN y no la clase MQADMIN.
2. Otorgue a cada usuario con el rol `MQWebUser` acceso a las colas del sistemas que son necesarios para utilizar la IBM MQ Console y la REST API.

   Para ello, tanto para SYSTEM.ADMIN.COMMAND.QUEUE como para SYSTEM.REST.REPLY.QUEUE, proporcione a cada usuario acceso UPDATE a las clases MQQUEUE o MXQUEUE, según si se utiliza o no seguridad que combina mayúsculas y minúsculas.

   Debe hacer esto en cada gestor de colas que el usuario administrará a través de la REST API, incluidos los gestores de colas remotos administrados a través de la pasarela deadministrative REST API.
3. Para permitir que un usuario con el rol `MQWebUser` administre los gestores de colas remotos, otorgue al usuario acceso UPDATE al perfil en la clase MQQUEUE o MXQUEUE, protegiendo la cola de transmisión utilizada para enviar mandatos a un gestor de colas remoto. Tenga en cuenta que debe proporcionar al usuario el acceso UPDATE en el gestor de colas de pasarela.

   En el gestor de colas remoto, otorgue acceso al mismo usuario para la colocación en la cola de transmisión utilizada para devolver mensajes de respuesta al gestor de colas de la pasarela.
4. Otorgue a los usuarios del rol `MQWebUser` acceso a cualquier otro recurso necesario para realizar las operaciones soportadas por la IBM MQ Console y la REST API.

   El acceso necesario para:

   - Realizar operaciones en la REST API, se describe en las secciones *Requisitos de seguridad* de los recursos de la REST API individuales.
   - Emitir mandatos mediante IBM MQ Console se describe en “IBM MQ Console - required command security profiles” en la página 235

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 Gestión de claves y certificados en AIX, Linux, and Windows

En AIX, Linux, and Windows, utilice los mandatos **runmqakm** y V 9.4.0 V 9.4.0 **runmqktool** para gestionar claves, certificados y solicitudes de certificados.

Acerca de esta tarea

El mandato **runmqakm** proporciona funciones similares a las de **gskitcapicmd**.

V 9.4.0 V 9.4.0 El mandato **runmqktool** proporciona funciones similares a las del programa de utilidad de gestión de certificados de Java **keytool** . Antes de utilizar los mandatos **runmqakm** o **runmqktool** , asegúrese de que las variables de entorno del sistema se hayan configurado correctamente ejecutando el mandato **setmqenv** .

El mandato **runmqktool** requiere que se instale el componente JRE de IBM MQ . Si este componente no está instalado, puede utilizar el mandato **runmqakm** en su lugar.

Si tiene que gestionar certificados TLS de una forma que sea compatible con el estándar FIPS, utilice el mandato **runmqakm**. Esto se debe a que el mandato **runmqakm** soporta un cifrado más potente.

Procedimiento

- Utilice los mandatos **runmqakm** y **runmqktool** para completar las acciones siguientes:
  - Cree un repositorio de claves CMS y PKCS #12 al que IBM MQ dé soporte.
  - Crear solicitudes de certificado.
  - Exportar certificados.
  - Importar certificados personales y certificados de CA.
  - Gestionar certificados autofirmados.
  - Crear, extraer y añadir claves secretas.

Información relacionada

Keytool

ALW Mandatos runmqakm y runmqktool en AIX, Linux, and Windows

En sistemas AIX, Linux, and Windows , utilice los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool) para gestionar claves y certificados.

Nota: V 9.4.0 V 9.4.0

A partir de IBM MQ 9.4.0, se eliminan los mandatos **runmqckm** y **strmqikm** . El mandato **runmqktool** se puede utilizar en lugar del mandato **runmqckm** para gestionar los repositorios de claves PKCS #12 y JKS. No hay sustitución para la GUI de **strmqikm** .

Los mandatos **runmqckm** y **runmqktool** tienen las siguientes diferencias importantes:

- El mandato **runmqktool** no da soporte a archivos de ocultación para almacenar contraseñas de repositorio de claves. La contraseña para acceder a un repositorio de claves debe proporcionarse siempre al mandato **runmqktool** cuando se ejecuta, ya sea como parámetro del mandato o en respuesta a una solicitud emitida por el mandato.
- El mandato **runmqktool** no da soporte a repositorios de claves CMS . Por lo tanto, para exportar un certificado de un JKS a un repositorio de claves CMS , debe completar los pasos siguientes:
  1. Utilice el mandato **runmqktool -importkeystore** para copiar el certificado del repositorio de claves JKS en un repositorio de claves PKCS #12 intermedio. Para obtener más información sobre la exportación de un certificado, consulte “Exportación de un certificado personal de un repositorio de claves en AIX, Linux, and Windows” en la página 561.
  2. Utilice el mandato **runmqakm -cert -import** para importar el certificado del repositorio de claves PKCS #12 intermedio al repositorio de claves CMS . Para obtener más información sobre la importación de un certificado, consulte “Importación de un certificado personal en un repositorio de claves en AIX, Linux, and Windows” en la página 563.

Los siguientes mandatos IBM MQ se pueden utilizar para gestionar claves y certificados:

runmqakm

- Proporciona funciones que son similares a las de **gskitcapicmd**.
- Da soporte a los repositorios de claves CMS y PKCS #12 .
- Da soporte a la creación de un archivo de ocultación para almacenar la contraseña del repositorio de claves cifrada.
- Certificado como compatible con FIPS 140-2, y se puede configurar para que funcione de forma compatible con FIPS con el parámetro **-fips** .

V 9.4.0 V 9.4.0 **runmqktool**

- Proporciona funciones que son similares a las del mandato Java **keytool** .
- Da soporte a los repositorios de claves PKCS #12, JKS y JCEKS.
- Requiere que el componente IBM MQ Java runtime environment (JRE) esté instalado.

Si necesita gestionar certificados de una forma que sea compatible con FIPS, utilice el mandato **runmqakm** .

Para obtener más información sobre el mandato **runmqakm** , consulte runmqakm.

V 9.4.0 V 9.4.0 Para obtener más información sobre el mandato **runmqktool** , consulte runmqktool.

Los temas de esta sección contienen ejemplos de cómo se utilizan estos mandatos para completar tareas comunes de gestión de certificados.

ALW Creación de un certificado personal autofirmado en AIX, Linux, and Windows

Siga este procedimiento para crear un certificado personal autofirmado en un repositorio de claves.

Nota: IBM MQ no da soporte a los algoritmos SHA-3 o SHA-5. Puede utilizar los nombres del algoritmo de firma digital SHA384WithRSA y SHA512WithRSA porque ambos algoritmos son miembros de la familia SHA-2.

Deprecated Los nombres de algoritmo de firma digital SHA3WithRSA y SHA5WithRSA están en desuso porque son una forma abreviada de SHA384WithRSA y SHA512WithRSA respectivamente.

Puede crear un certificado autofirmado utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Para obtener más información sobre por qué utilizar certificados autofirmados, consulte Utilización de certificados autofirmados para la autenticación mutua de dos gestores de colas.

No todos los certificados digitales se pueden utilizar con todas las CipherSpecs. Asegúrese de crear un certificado que sea compatible con las CipherSpecs que utilice. IBM MQ da soporte a tres tipos distintos de CipherSpec. Para obtener más información, consulte “Interoperatividad de Elliptic Curve y CipherSpecs RSA” en la página 50.

Para utilizar las CipherSpecs de tipo 1 (aquellas con nombres que empiezan por ECDHE_ECDSA_), debe utilizar el mandato **runmqakm** para crear el certificado y debe especificar un parámetro de algoritmo de firma Elliptic Curve ECDSA. Por ejemplo, especificando el parámetro **-sig_alg** EC_ecdsa_with_SHA384.

Utilización de runmqakm

Emita el mandato siguiente para crear un certificado personal autofirmado con el mandato **runmqakm** :

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

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves. El repositorio de claves ya debe existir.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta de certificado. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

La etiqueta de un certificado TLS que utiliza IBM MQ es el valor del atributo **CERTLABL** si está establecido, o el valor predeterminado ibmwebspheremq con el nombre del gestor de colas o el ID de usuario IBM MQ MQI client añadido, todo en minúsculas. Para obtener más información, consulte “Etiquetas de certificados digitales, descripción de los requisitos” en la página 27.

-dn *nombre_distinguido*
Especifica el nombre distinguido X.500 especificado entre comillas dobles. Se necesita al menos un atributo en el nombre distinguido. Puede proporcionar varios atributos OU y DC.

Nota: El mandato **runmqakm** hace referencia al atributo de código postal como POSTALCODE, no PC. Especifique siempre POSTALCODE en el parámetro **-dn** cuando utilice el mandato **runmqakm** para solicitar certificados con un código postal.

-size *tamaño_clave*
Especifica el tamaño de clave. El valor puede ser 512, 1024o 2048.

-x509version *versión*
Versión del certificado X.509 que se debe crear. El valor puede ser 1, 2 o 3. El valor predeterminado es 3.

-expire *Días*
Tiempo de caducidad del certificado en días. El valor predeterminado para un certificado es 365 días.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Solo se utiliza el componente FIPS IBM Crypto for C (ICC) y este componente debe inicializarse correctamente en modalidad FIPS. Cuando está en modalidad FIPS, el componente ICC utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

-sig_alg
Especifica el algoritmo de hash que se utiliza cuando se crea el certificado. Este algoritmo de hash se utiliza para crear la firma asociada con el certificado. El valor puede 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` o `EC_ecdsa_with_SHA512`.

El valor predeterminado es `SHA1WithRSA`.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

Utilización de runmqktool

V 9.4.0 V 9.4.0

Emita el mandato siguiente para crear un certificado personal autofirmado con el mandato **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
```

donde:

-keystore *nombre_archivo*
Especifica el nombre del repositorio de claves. El repositorio de claves se crea si no existe.

-storepass *contraseña*
Especifica la contraseña del repositorio de claves.

-storetype *tipo_tienda*
Especifica el tipo de repositorio de claves.

-alias *etiqueta*
Especifica la etiqueta de certificado. La etiqueta de certificado se convierte a minúsculas.

-dname *nombre_distinguido*
Especifica el nombre distinguido X.500 para el certificado entre comillas dobles.

-validez *días*
Especifica el número de días durante los cuales el certificado es válido.

-keyalg *algoritmo_clave*
Especifica el algoritmo que se utiliza para crear el par de claves.

-keysize *tamaño_clave*
Especifica el tamaño de clave.

-sigalg *algoritmo_firma*
Especifica el algoritmo que se utiliza para firmar el certificado. Para obtener más información sobre los algoritmos de firma que se pueden especificar, consulte Algoritmos de firma.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte genkeypair.

ALW Solicitud de un certificado personal en AIX, Linux, and Windows

Siga este procedimiento para crear una solicitud de un certificado personal.

Nota: IBM MQ no da soporte a los algoritmos SHA-3 o SHA-5. Puede utilizar los nombres del algoritmo de firma digital SHA384WithRSA y SHA512WithRSA porque ambos algoritmos son miembros de la familia SHA-2.

Deprecated Los nombres de algoritmo de firma digital SHA3WithRSA y SHA5WithRSA están en desuso porque son una forma abreviada de SHA384WithRSA y SHA512WithRSA respectivamente.

Puede solicitar un certificado personal utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

No todos los certificados digitales se pueden utilizar con todas las CipherSpecs. Asegúrese de crear un certificado que sea compatible con las CipherSpecs que utilice. IBM MQ da soporte a tres tipos distintos de CipherSpec. Para obtener más información, consulte “Interoperatividad de Elliptic Curve y CipherSpecs RSA” en la página 50.

Para utilizar las CipherSpecs de tipo 1 (aquellas con nombres que empiezan por ECDHE_ECDSA_), debe utilizar el mandato **runmqakm** para crear el certificado y debe especificar un parámetro de algoritmo de firma Elliptic Curve ECDSA. Por ejemplo, especificando el parámetro **-sig_alg** EC_ecdsa_with_SHA384.

Si utiliza hardware de cifrado, consulte “Solicitud de un certificado personal para el hardware PKCS #11” en la página 573.

Utilización de runmqakm

Emita el mandato siguiente para crear una solicitud de certificado con el mandato **runmqakm** :

```
runmqakm -certreq -create -db filename -pw password -label label
         -dn distinguished_name -size key_size
         -file filename -fips -sig_alg algorithm
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo de un repositorio de claves. El repositorio de claves ya debe existir.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta de certificado. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

La etiqueta de un certificado TLS que utiliza IBM MQ es el valor del atributo **CERTLABL** si está establecido, o el valor predeterminado ibmwebspheremq con el nombre del gestor de colas o el ID de usuario IBM MQ MQI client añadido, todo en minúsculas. Para obtener más información, consulte “Etiquetas de certificados digitales, descripción de los requisitos” en la página 27.

-dn *nombre_distinguido*
Especifica el nombre distinguido X.500 especificado entre comillas dobles. Se necesita al menos un atributo en el nombre distinguido. Puede proporcionar varios atributos OU y DC.

Nota: El mandato **runmqakm** hace referencia al atributo de código postal como POSTALCODE, no PC. Especifique siempre POSTALCODE en el parámetro **-dn** cuando utilice el mandato **runmqakm** para solicitar certificados con un código postal.

-size *tamaño_clave*
Especifica el tamaño de clave. El valor puede ser 512, 1024o 2048.

-file *nombrearchivo*
Especifica el nombre de archivo para la solicitud de certificado.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que están validados por FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

-sig_alg
Especifica el algoritmo de hash que se utiliza cuando se crea la solicitud de certificado. Este algoritmo de hash se utiliza para crear la firma asociada con la solicitud de certificado. El valor puede 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` o `EC_ecdsa_with_SHA512`.

El valor predeterminado es `SHA1WithRSA`.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm-certreq .

Utilización de runmqktool

V 9.4.0 V 9.4.0

Para poder crear una solicitud de certificado con el mandato **runmqktool** , debe generar un par de claves utilizando el mandato **runmqktool -genkeypair** . Para obtener más información sobre el mandato **runmqktool -genkeypair** , consulte “Creación de un certificado personal autofirmado en AIX, Linux, and Windows” en la página 551.

Emita el mandato siguiente para crear una solicitud de certificado con el mandato **runmqktool** :

```
runmqktool -certreq -keystore filename -storepass password -alias label
           -file filename
```

donde:

-keystore *nombre_archivo*
Especifica el nombre del repositorio de claves.

-storepass *contraseña*
Especifica la contraseña del repositorio de claves.

-alias *etiqueta*
Especifica la etiqueta de certificado. Esta es la etiqueta de certificado que se ha especificado al generar el par de claves. La etiqueta del certificado no distingue entre mayúsculas y minúsculas.

-file *nombrearchivo*
Especifica el nombre de archivo para la solicitud de certificado.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte certreq.

Qué hacer a continuación

Envíe una solicitud de certificado a una CA. Cuando reciba el certificado firmado de la CA, añada el certificado firmado al repositorio de claves. Para obtener más información, consulte “Recepción de certificados personales en un repositorio de claves en AIX, Linux, and Windows” en la página 556.

ALW Renovación de un certificado personal existente en AIX, Linux, and Windows

Un certificado personal tiene una fecha de caducidad, tras la cuál ya no se puede utilizar el certificado. Siga este procedimiento para renovar un certificado personal antes de que caduque.

Puede renovar un certificado personal utilizando el mandato **runmqakm** (GSKCapiCmd).

Si tiene un requisito de utilizar tamaños de clave mayores para sus certificados personales, no puede renovar un certificado existente. Debe sustituir la clave existente siguiendo los pasos descritos en "Solicitud de un certificado personal en AIX, Linux, and Windows" en la página 554 para crear una nueva solicitud de certificado que utilice los tamaños de clave que necesite.

Utilización de runmqakm

Emita el mandato siguiente para crear una solicitud de certificado para renovar un certificado personal con el mandato **runmqakm** :

```
runmqakm -certreq -recreate -db filename -pw password
         -label label -target filename
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta de certificado. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-target *nombre_archivo*
Especifica el nombre de archivo para la solicitud de certificado.

Qué hacer a continuación

Envíe una solicitud de certificado a una CA. Cuando reciba el certificado firmado de la CA, añada el certificado firmado al repositorio de claves. Para obtener más información, consulte "Recepción de certificados personales en un repositorio de claves en AIX, Linux, and Windows" en la página 556.

ALW Recepción de certificados personales en un repositorio de claves en AIX, Linux, and Windows

Utilice este procedimiento para recibir un certificado personal en el repositorio de claves.

Después de que la entidad emisora de certificados (CA) le envíe un nuevo certificado personal, añádalo al repositorio de claves desde el que ha generado la nueva solicitud de certificado. Si la CA envía el certificado como parte de un mensaje de correo electrónico, copie el certificado en un archivo aparte.

Antes de añadir el certificado personal firmado por CA al repositorio de claves, realice los pasos de "Adición de un certificado de CA, o la parte pública de un certificado de confianza, a un repositorio de claves en AIX, Linux, and Windows" en la página 560 para añadir el certificado de CA al repositorio de claves.

Puede recibir un certificado personal en un repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Si utiliza hardware de cifrado, consulte el apartado "Recepción de un certificado personal en el hardware PKCS #11" en la página 574.

Utilización de `runmqakm`

Emita el mandato siguiente para añadir un certificado personal a un repositorio de claves con el mandato **`runmqakm`** :

```
runmqakm -cert -receive -file filename -format format
         -db filename -pw password -fips
```

donde:

-file *nombrearchivo*
Especifica el nombre de archivo completo del certificado personal.

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves. El repositorio de claves ya debe existir y debe ser el mismo repositorio donde ha creado la solicitud de certificado.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-format *formato*
Especifica el formato del certificado. El valor puede ser `ascii` para datos ASCII codificados con Base64 o bien `binary` para datos DER binarios. El valor predeterminado es `ascii`.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **`runmqakm`** falla.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

Utilización de `runmqktool`

V 9.4.0 V 9.4.0

Emita el mandato siguiente para añadir un certificado personal a un repositorio de claves con el mandato **`runmqktool`** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

donde:

-keystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves. El repositorio de claves ya debe existir y debe ser el mismo repositorio donde ha creado la solicitud de certificado.

-storepass *contraseña*
Especifica la contraseña del repositorio de claves.

-alias *etiqueta*
Especifica la etiqueta del certificado que se ha utilizado para crear la solicitud de certificado. La etiqueta de certificado se convierte a minúsculas.

-file *nombrearchivo*
Especifica el nombre de archivo completo del certificado personal.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte importcert.

Qué hacer a continuación

Si el certificado se añade al repositorio de claves TLS del gestor de colas, emita el mandato MQSC **`REFRESH SECURITY TYPE(SSL)`** para renovar la memoria caché del repositorio de claves TLS del gestor de colas.

ALW Extracción de un certificado de CA de un repositorio de claves en AIX, Linux, and Windows

Siga este procedimiento para extraer un certificado de entidad emisora de certificados (CA) de un repositorio de claves.

Puede extraer un certificado de CA de un repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Utilización de runmqakm

Emita el mandato siguiente para extraer un certificado de CA con el mandato **runmqakm** :

```
runmqakm -cert -extract -db filename -pw password -label label
         -target filename -format format -fips
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta del certificado de CA. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-target *nombre_archivo*
Especifica el nombre de archivo completo del archivo de destino.

-format *formato*
Especifica el formato del certificado. El valor puede ser `ascii` para datos ASCII codificados con Base64 o bien `binary` para datos DER binarios. El valor predeterminado es `ascii`.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

Utilización de runmqktool

V 9.4.0 V 9.4.0

Emita el mandato siguiente para extraer un certificado de CA con el mandato **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

donde:

-keystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves.

-storepass *contraseña*
Especifica la contraseña del repositorio de claves.

-alias *etiqueta*
Especifica la etiqueta del certificado de CA. La etiqueta del certificado no distingue entre mayúsculas y minúsculas.

-file *nombrearchivo*
Especifica el nombre de archivo completo del archivo de destino.

-rfc
Especifica que el archivo de salida está en formato ASCII Base64-encoded , tal como lo define el estándar RFC 1421 de Internet. Si no se especifica esta opción, el archivo de salida está en formato binario.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte exportcert.

ALW Extracción de la parte pública de un certificado autofirmado de un repositorio de claves en AIX, Linux, and Windows

Siga este procedimiento para extraer la parte pública de un certificado autofirmado de un repositorio de claves.

Puede extraer la parte pública de un certificado de un repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Utilización de runmqakm

Emita el mandato siguiente para extraer la parte pública de un certificado autofirmado con el mandato **runmqakm** :

```
runmqakm -cert -extract -db filename -pw password -label label
           -target filename -format format -fips
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta del certificado de CA. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-target *nombre_archivo*
Especifica el nombre de archivo completo del archivo de destino.

-format *formato*
Especifica el formato del certificado. El valor puede ser `ascii` para datos ASCII codificados con Base64 o bien `binary` para datos DER binarios. El valor predeterminado es `ascii`.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert .

Utilización de runmqktool

V 9.4.0 V 9.4.0

Emita el mandato siguiente para extraer la parte pública de un certificado autofirmado con el mandato **runmqktool** :

```
runmqktool -exportcert -keystore filename -storepass filename -alias label
           -file filename -rfc
```

donde:

-keystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves.

-storepass *contraseña*
Especifica la contraseña del repositorio de claves.

-alias *etiqueta*
Especifica la etiqueta del certificado de CA. La etiqueta del certificado no distingue entre mayúsculas y minúsculas.

-file *nombrearchivo*
Especifica el nombre de archivo completo del archivo de destino.

-rfc
Especifica que el archivo de salida está en formato ASCII Base64-encoded , tal como lo define el estándar RFC 1421 de Internet. Si no se especifica esta opción, el archivo de salida está en formato binario.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte exportcert.

ALW Adición de un certificado de CA, o la parte pública de un certificado de confianza, a un repositorio de claves en AIX, Linux, and Windows

Siga este procedimiento para añadir un certificado de CA o la parte pública de un certificado de confianza a un repositorio de claves.

Puede añadir un certificado de CA, o la parte pública de un certificado de confianza, a un repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Si el certificado que desea añadir se encuentra en una cadena de certificados, también debe añadir todos los certificados que están por encima suyo en la cadena. Debe añadir los certificados en orden estrictamente descendente, empezando por el raíz, seguido del certificado de CA inmediatamente debajo de éste en la cadena, y así sucesivamente.

Nota:

- Asegúrese de que el certificado esté en codificación ASCII (UTF-8) o binaria (DER).
- Debido a una restricción en el mandato IBM Java 8 **keytool** , **runmqktool** no puede importar certificados en formato de codificación imprimible (también conocido como codificación Base64) tal como se define en la RFC 1421 de Internet si el archivo contiene comentarios. Para importar un certificado en formato de codificación imprimible, elimine todos los comentarios del archivo. El archivo debe empezar con una serie que empiece por "----- BEGIN", y finalizar con una serie que empiece por "----- END".

Utilización de runmqakm

Emita el mandato siguiente para añadir un certificado de confianza a un repositorio de claves con el mandato **runmqakm** :

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

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves. El repositorio de claves ya debe existir.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta de certificado. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-file *nombrearchivo*
Especifica el nombre del archivo que contiene el certificado.

-format *ascii*
Especifica el formato del certificado. El valor puede ser `ascii` para datos ASCII codificados con Base64 o bien `binary` para datos DER binarios. El valor predeterminado es `ascii`.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

Utilización de `runmqktool`

V 9.4.0 V 9.4.0

Emita el mandato siguiente para añadir un certificado de confianza a un repositorio de claves con el mandato **runmqktool** :

```
runmqktool -importcert -keystore filename -storepass password
           -alias label -file filename
```

donde:

-keystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves. El repositorio de claves se crea si no existe.

-storepass *contraseña*
Especifica la contraseña del repositorio de claves.

-alias *etiqueta*
Especifica la etiqueta de certificado. La etiqueta de certificado se convierte a minúsculas.

-file *nombrearchivo*
Especifica el nombre de archivo completo del certificado personal.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte importcert.

ALW Exportación de un certificado personal de un repositorio de claves en AIX, Linux, and Windows

Siga este procedimiento para exportar un certificado personal desde un repositorio de claves.

La exportación de un certificado copia el certificado y sus claves públicas y privadas asociadas en otro repositorio de claves.

Puede exportar un certificado desde un repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Utilización de `runmqakm`

Emita el mandato siguiente para exportar un certificado con el mandato **`runmqakm`** :

```
runmqakm -cert -export -db filename -pw password -label label
         -target filename -target_pw password -target_type type
         -encryption strength -fips
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves que contiene el certificado.

-pw *contraseña*
Especifica la contraseña del repositorio de claves que contiene el certificado.

-label *etiqueta*
Especifica la etiqueta del certificado que se va a exportar. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-target *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves de destino. El repositorio de claves se crea si no existe.

-target_pw *contraseña*
Especifica la contraseña del repositorio de claves de destino.

-target_type *tipo*
Especifica el tipo del repositorio de claves de destino. El valor puede ser `cms` o `pkcs12`. El valor predeterminado es `cms`.

-encryption *fortaleza*
Especifica la fuerza del cifrado que se utiliza en el mandato de exportación de certificados. El valor puede ser `STRONG` o `WEAK`. El valor predeterminado es `STRONG`.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **`runmqakm`** falla.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

Utilización de `runmqktool`

V 9.4.0 V 9.4.0

Emita el mandato siguiente para exportar un certificado con el mandato **`runmqktool`** :

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

donde:

-srckeystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves que contiene el certificado.

-srcstorepass *contraseña*
Especifica la contraseña del repositorio de claves que contiene el certificado.

-destkeystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves de destino. El repositorio de claves se crea si no existe.

-deststorepass *contraseña*
Especifica la contraseña del repositorio de claves de destino.

-destkeypass *contraseña*
Especifica la contraseña para proteger la clave en el repositorio de claves de destino. Si no se especifica este parámetro, la clave se protege con la contraseña que se utiliza para proteger la clave en el repositorio de claves de origen.

-deststoretype *tipo*
Especifica el tipo del repositorio de claves de destino.

-srcalias *etiqueta*
Especifica la etiqueta del certificado que se va a exportar. La etiqueta del certificado no distingue entre mayúsculas y minúsculas.

-destalias *etiqueta*
Especifica la etiqueta del certificado en el repositorio de claves de destino. Si no se especifica este parámetro, se asigna la misma etiqueta al certificado que en el repositorio de claves de origen.

La etiqueta de certificado se convierte a minúsculas.

-file *nombrearchivo*
Especifica el nombre de archivo completo del archivo de destino.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte importkeystore.

ALW Importación de un certificado personal en un repositorio de claves en AIX, Linux, and Windows

Siga este procedimiento para importar un certificado personal en un repositorio de claves.

La importación de un certificado copia el certificado y sus claves públicas y privadas asociadas de un repositorio de claves a otro repositorio de claves.

Antes de importar un certificado personal a un repositorio de claves, primero debe añadir la cadena válida completa de emisión de certificados de CA al repositorio de claves. Para obtener más información, consulte “Adición de un certificado de CA, o la parte pública de un certificado de confianza, a un repositorio de claves en AIX, Linux, and Windows” en la página 560.

Puede importar un certificado a un repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Utilización de runmqakm

Emita el mandato siguiente para importar un certificado con el mandato **runmqakm** :

```
runmqakm -cert -import -file filename -pw password -type type
         -target filename -target_pw password -target_type type
         -label label -new_label label -fips
```

donde:

-file *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves que contiene el certificado.

-pw *contraseña*
Especifica la contraseña del repositorio de claves que contiene el certificado.

-type *tipo*
Especifica el tipo del repositorio de claves que contiene el certificado. El valor puede ser cms o pkcs12. El valor predeterminado es cms.

-target *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves de destino. El repositorio de claves se crea si no existe.

-target_pw *contraseña*
Especifica la contraseña del repositorio de claves de destino.

-target_type *tipo*
Especifica el tipo del repositorio de claves de destino. El valor puede ser `cms` o `pkcs12`. El valor predeterminado es `cms`.

-label *etiqueta*
Especifica la etiqueta del certificado que se va a importar desde el repositorio de claves de origen. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-new_label *etiqueta*
Especifica la etiqueta que se asigna al certificado en el repositorio de claves de destino. Si no se especifica este parámetro, se asigna la misma etiqueta al certificado que en el repositorio de claves de origen.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert .

Utilización de runmqktool

V 9.4.0 V 9.4.0

Emita el mandato siguiente para importar un certificado con el mandato **runmqktool** :

```
runmqktool -importkeystore -srckeystore filename -srcstorepass password
           -destkeystore filename -deststoretype type
           -deststorepass password -destkeypass password
           -srcalias label -destalias label
```

donde:

-srckeystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves que contiene el certificado.

-srcstorepass *contraseña*
Especifica la contraseña del repositorio de claves que contiene el certificado.

-destkeystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves de destino. El repositorio de claves se crea si no existe.

-deststorepass *contraseña*
Especifica la contraseña del repositorio de claves de destino.

-destkeypass *contraseña*
Especifica la contraseña para proteger la clave en el repositorio de claves de destino. Si no se especifica este parámetro, la clave se protege con la contraseña que se utiliza para proteger la clave en el repositorio de claves de origen.

Nota: Para un repositorio de claves PKCS #12 , la clave debe estar protegida con la misma contraseña que el repositorio de claves de destino.

-deststoretype *tipo*
Especifica el tipo del repositorio de claves de destino.

-srcalias *etiqueta*
Especifica la etiqueta del certificado en el repositorio de claves de origen. La etiqueta del certificado no distingue entre mayúsculas y minúsculas.

-destalias *etiqueta*
Especifica la etiqueta del certificado en el repositorio de claves de destino. Si no se especifica este parámetro, se asigna la misma etiqueta al certificado que en el repositorio de claves de origen.

La etiqueta de certificado se convierte a minúsculas.

-file *nombrearchivo*
Especifica el nombre de archivo completo del archivo de destino.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte importkeystore.

ALW Importación de un certificado personal desde un archivo Microsoft.pfx

Siga este procedimiento para importar un certificado desde unMicrosoft archivo .pfx enAIX, Linux, and Windows .

Un archivo .pfx puede contener dos certificados relativos a la misma clave. Uno es un certificado personal o de sitio que contiene una clave pública y privada. El otro es un certificado de CA (firmante) que contiene sólo una clave pública. Estos certificados no pueden coexistir en el mismo repositorio de claves CMS , por lo que solo se puede importar uno de ellos.

La etiqueta de certificado se adjunta sólo al certificado de firmante. El certificado personal se identifica mediante un identificador de usuario único (UUID) generado por el sistema. Siga este procedimiento para importar un certificado personal desde un archivo .pfx y establecer la etiqueta de certificado personal en la etiqueta asignada al certificado CA en el archivo .pfx. Los certificados de CA que emiten ya deben añadirse a la base de datos de claves de destino.

Utilización de `runmqakm`

Emita el mandato siguiente para importar un certificado desde un archivo .pfx con el mandato **`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
```

donde:

-file *nombrearchivo*
Especifica el nombre completo del archivo .pfx.

-pw *contraseña*
Especifica la contraseña para el archivo .pfx.

-type pkcs12
Especifica el tipo del repositorio de claves.

-target *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves de destino. El repositorio de claves se crea si no existe.

-target_pw *contraseña*
Especifica la contraseña del repositorio de claves de destino.

-target_type *tipo*
Especifica el tipo del repositorio de claves de destino. El valor puede ser `cms` o `pkcs12`. El valor predeterminado es `cms`.

-label *etiqueta*
Especifica la etiqueta del certificado que se va a importar desde el repositorio de claves de origen. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-new_label *etiqueta*
Especifica la etiqueta que se asigna al certificado en el repositorio de claves de destino. Si no se especifica este parámetro, se asigna la misma etiqueta al certificado que en el repositorio de claves de origen.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

-pfx
Indica que el repositorio de claves de origen utiliza el formato PFX.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

ALW Importación de un certificado personal desde un archivo PKCS #7

Siga este procedimiento para importar un certificado de un archivo PKCS #7 en AIX, Linux, and Windows.

Utilice el mandato **runmqakm** para importar certificados de un archivo PKCS #7 en AIX, Linux, and Windows.

Adición de un certificado de CA o la parte pública de un certificado de confianza

Emita el mandato siguiente para añadir un certificado de CA, o la parte pública de un certificado de confianza, desde un archivo PKCS #7 :

```
runmqakm -cert -add -db filename -pw password -type type
         -label label -file filename
```

donde:

-db *nombrearchivo*
Especifica el nombre completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-type *tipo*
Especifica el tipo del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta del certificado que se va a añadir. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

La etiqueta se asigna al primer certificado que se añade. Todos los demás certificados, si los hay, utilizan el nombre de asunto como etiqueta.

-file *nombrearchivo*
Especifica el nombre completo del archivo PKCS #7 .

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

Importación de un certificado personal

Emita el mandato siguiente para importar un certificado personal desde un archivo PKCS #7 :

```
runmqakm -cert -import -file filename -pw password -type pkcs7
         -target filename -target_pw password -target_type type
         -label label -new_label label
```

donde:

-file *nombrearchivo*
Especifica el nombre completo del archivo PKCS #7 .

-pw *contraseña*
Especifica la contraseña para el archivo PKCS #7 .

-type pkcs7
Especifica el tipo del archivo PKCS #7 .

-target *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves de destino. El repositorio de claves se crea si no existe.

-target_pw *contraseña*
Especifica la contraseña del repositorio de claves de destino.

-target_type *tipo*
Especifica el tipo del repositorio de claves de destino. El valor puede ser `cms` o `pkcs12`. El valor predeterminado es `cms`.

-label *etiqueta*
Especifica la etiqueta del certificado que se va a importar desde el archivo PKCS #7 . La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-new_label *etiqueta*
Especifica la etiqueta que se asigna al certificado en el repositorio de claves de destino. Si no se especifica este parámetro, se asigna la misma etiqueta al certificado que en el repositorio de claves de origen.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

ALW Listado de los certificados en un repositorio de claves en AIX, Linux, and Windows

Utilice este procedimiento para listar los certificados que están en un repositorio de claves.

Puede visualizar información sobre los certificados que están en un repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool).

Utilización de runmqakm

- Emita el mandato siguiente para listar las etiquetas de los certificados en un repositorio de claves con el mandato **runmqakm** :

```
runmqakm -cert -list -db filename -pw password
```

- Emita el mandato siguiente para listar los detalles de un certificado en un repositorio de claves con el mandato **runmqakm** :

```
runmqakm -cert -details -showOID -db filename -pw password
         -label label
```

donde:

-file *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta del certificado que se va a listar. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert.

Utilización de runmqktool

V 9.4.0 V 9.4.0

- Emita el mandato siguiente para listar las etiquetas de los certificados en un repositorio de claves con el mandato **runmqktool** :

```
runmqktool -list -keystore filename -storepass password
```

- Emita el mandato siguiente para listar los detalles de un certificado en un repositorio de claves con el mandato **runmqktool** :

```
runmqktool -list -keystore filename -storepass password -alias label -v
```

donde:

-keystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves.

-storepass *contraseña*
Especifica la contraseña del repositorio de claves.

-alias *etiqueta*
Especifica la etiqueta del certificado que se va a listar. La etiqueta del certificado no distingue entre mayúsculas y minúsculas.

-v
Solicita una salida detallada que incluye los detalles del certificado.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte lista.

ALW Supresión de un certificado de un repositorio de claves en AIX, Linux, and Windows

Utilice este procedimiento para suprimir un certificado personal o de CA de un repositorio de claves.

Puede suprimir un certificado de un repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool). Si tiene que gestionar certificados SSL o TLS de una forma que sea compatible con FIPS, utilice el mandato **runmqakm**.

Utilización de runmqakm

Emita el mandato siguiente para suprimir un certificado con el mandato **runmqakm** :

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

donde:

-file *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta del certificado que se va a suprimir. La etiqueta del certificado distingue entre mayúsculas y minúsculas.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que se han validado con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -cert .

Utilización de runmqktool

V 9.4.0 V 9.4.0

Emita el mandato siguiente para suprimir un certificado con el mandato **runmqktool** :

```
runmqktool -delete -keystore filename -storepass password -alias label
```

donde:

-keystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves.

-storepass *contraseña*
Especifica la contraseña del repositorio de claves.

-alias *etiqueta*
Especifica la etiqueta del certificado que se va a suprimir. La etiqueta del certificado no distingue entre mayúsculas y minúsculas.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte delete.

ALW Conversión de un repositorio de claves en AIX, Linux, and Windows

Utilice este procedimiento para convertir un repositorio de claves a un tipo diferente.

Puede convertir una contraseña de repositorio de claves a un tipo diferente utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool).

Utilización de runmqakm

Emita el mandato siguiente para convertir un repositorio de claves con el mandato **runmqakm** :

```
runmqakm -keydb -convert -db filename -pw password
         -new_db filename -new_pw password
         -old_format type -new_format type
```

donde:

-file *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-new_db *nombre_archivo*
Especifica el nombre de archivo completo del nuevo repositorio de claves.

-new_pw *contraseña*
Especifica la contraseña para el nuevo repositorio de claves.

-old_format *tipo*
Especifica el tipo actual del repositorio de claves. Se pueden especificar los siguientes valores:

- pkcs12
- cms

-new_format *tipo*
Especifica el nuevo tipo del repositorio de claves. Se pueden especificar los siguientes valores:

- pkcs12
- cms

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -keydb.

Utilización de runmqktool

V 9.4.0 V 9.4.0

Emita el mandato siguiente para convertir un repositorio de claves con el mandato **runmqktool** :

```
runmqktool -importkeystore -srckeystore filename -destkeystore filename
           -srcstoretype type -deststoretype type
           -srcstorepass password -deststorepass password
```

donde:

-all
Especifica que la contraseña también se cambia para todas las entradas que están protegidas con la misma contraseña que el repositorio de claves.

-keystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves.

-destkeystore *nombre_archivo*
Especifica el nombre de archivo completo del nuevo repositorio de claves.

-srcstoretype *tipo*
Especifica el tipo de repositorio de claves.

-deststoretype *tipo*
Especifica el nuevo tipo de repositorio de claves.

-srcstorepass *contraseña*
Especifica la contraseña del repositorio de claves.

-deststorepass *contraseña*
Especifica la contraseña para el nuevo repositorio de claves.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte importkeystore.

ALW Cambio de la contraseña del repositorio de claves en AIX, Linux, and Windows

Utilice este procedimiento para cambiar la contraseña del repositorio de claves.

Puede cambiar la contraseña del repositorio de claves utilizando los mandatos **runmqakm** (GSKCapiCmd) o **runmqktool** (keytool).

Nota:

- V 9.4.0 V 9.4.0 El mandato **runmqktool** permite cambiar la contraseña del repositorio de claves independientemente de las contraseñas que protegen las claves privadas y secretas individuales. Para los repositorios de claves PKCS #12 , la contraseña del repositorio de claves y las contraseñas que protegen todas las claves del repositorio de claves deben ser iguales. Si se utiliza el mandato **runmqktool** para cambiar la contraseña del repositorio de claves, asegúrese de que se ha especificado el parámetro **-all** para que también se cambien las contraseñas clave.
- Si la contraseña del repositorio de claves no se almacena en un archivo de ocultación, también debe cambiar la contraseña almacenada en la configuración del gestor de colas o en cualquier aplicación IBM MQ client que acceda al repositorio de claves. Para obtener más información, consulte “Suministro de la contraseña del repositorio de claves para un gestor de colas en AIX, Linux, and Windows” en la página 306 y “Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows” en la página 308.

Utilización de runmqakm

Emita el mandato siguiente para cambiar la contraseña del repositorio de claves con el mandato **runmqakm** :

```
runmqakm -keydb -changepw -db filename -pw password -new_pw password -stash
```

donde:

-file *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña actual para el repositorio de claves.

-new_pw *contraseña*
Especifica la nueva contraseña para el repositorio de claves.

-stash
Opcional. Especifique esta opción para almacenar la nueva contraseña del repositorio de claves en un archivo de ocultación. No es necesario almacenar la contraseña en un archivo de ocultación si cifra la contraseña utilizando en su lugar el sistema de protección de contraseñas de IBM MQ .

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte ejecutarmqakm-keydb .

Utilización de `runmqktool`

V 9.4.0 V 9.4.0

Emita el mandato siguiente para cambiar la contraseña del repositorio de claves con el mandato **`runmqktool`** :

```
runmqktool -storepasswd -all -keystore filename -storepass password
           -new password
```

donde:

-all
Especifica que la contraseña también se cambia para todas las entradas que están protegidas con la misma contraseña que el repositorio de claves.

-keystore *nombre_archivo*
Especifica el nombre de archivo completo del repositorio de claves.

-storepass *contraseña*
Especifica la contraseña actual para el repositorio de claves.

-new *contraseña*
Especifica la nueva contraseña para el repositorio de claves.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte storepasswd.

ALW Gestión de claves secretas en AIX, Linux, and Windows

Siga este procedimiento para gestionar claves secretas en un repositorio de claves.

Puede gestionar claves secretas utilizando el mandato **`runmqakm`** (GSKCapiCmd). Las claves secretas que se generan utilizando el mandato **`runmqktool`** (keytool) no se pueden utilizar con IBM MQ.

Creación de una clave secreta

Emita el mandato siguiente para crear una clave secreta aleatoria con el mandato **`runmqakm`** :

```
runmqakm -secretkey -create -db filename -pw password
         -label label -size key_size
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves. El repositorio de claves ya debe existir.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta que se adjunta a la clave.

-size *tamaño_clave*
Especifica el tamaño de clave en bytes.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -clave secreta .

Extracción de una clave secreta

Emita el mandato siguiente para extraer una clave secreta con el mandato **runmqakm** :

```
runmqakm -secretkey -extract -db filename -pw password
         -label label -target filename -format format
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves. El repositorio de claves ya debe existir.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta de la clave que se va a extraer.

-target *nombre_archivo*
Especifica el nombre de archivo completo del archivo de destino.

-format *formato*
Especifica el formato de la clave en el archivo de destino. El valor puede ser `ascii` para ASCII Base64-encoded o `binary` para una copia binaria de la clave. El valor predeterminado es `ascii`.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -clave secreta .

Adición de una clave secreta

Emita el mandato siguiente para extraer una clave secreta con el mandato **runmqakm** :

```
runmqakm -secretkey -add -db filename -pw password
         -label label -file filename -format format
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves. El repositorio de claves ya debe existir.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-label *etiqueta*
Especifica la etiqueta que se adjunta a la clave.

-file *nombrearchivo*
Especifica el nombre del archivo que contiene la clave.

-format *formato*
Especifica el formato de la clave. El valor puede ser `ascii` para ASCII Base64-encoded o `binary` para datos binarios. El valor predeterminado es `ascii`.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -clave secreta .

ALW Gestión de certificados en el hardware PKCS #11

Puede gestionar certificados digitales en el hardware de cifrado que da soporte a la interfaz PKCS #11.

Debe crear un repositorio de claves para preparar el entorno de IBM MQ , aunque no tenga previsto almacenar ningún certificado en él, pero almacenará todos los certificados en el hardware criptográfico. Es necesario un repositorio de claves para que el gestor de colas haga referencia a su atributo **SSLKEYR** , o para que la aplicación cliente haga referencia a la variable de entorno MQSSLKEYR. Este repositorio de claves también es necesario si está creando una solicitud de certificado.

Cree el repositorio de claves utilizando el mandato **runmqakm** (GSKCapiCmd).

Emita el mandato siguiente para crear un repositorio de claves con el mandato **runmqakm** :

```
runmqakm -keydb -create -db filename -pw password -type type -stash
```

donde:

-db *nombrearchivo*
Especifica el nombre de archivo completo del repositorio de claves.

-pw *contraseña*
Especifica la contraseña del repositorio de claves.

-type *tipo*
Especifica el tipo de base de datos. El valor debe ser `cms` o `pkcs12` para un repositorio de claves utilizado por IBM MQ.

-stash
Opcional. Si se especifica, la contraseña del repositorio de claves cifrada se guarda en un archivo.

ALW *Solicitud de un certificado personal para el hardware PKCS #11*

Utilice este procedimiento para solicitar un certificado personal para un gestor de colas o un IBM MQ MQI client con el hardware de cifrado.

Nota: IBM MQ no da soporte a los algoritmos SHA-3 o SHA-5. Puede utilizar los nombres del algoritmo de firma digital SHA384WithRSA y SHA512WithRSA porque ambos algoritmos son miembros de la familia SHA-2.

Deprecated Los nombres de algoritmo de firma digital SHA3WithRSA y SHA5WithRSA están en desuso porque son una forma abreviada de SHA384WithRSA y SHA512WithRSA respectivamente.

Antes de crear una solicitud de certificado en el hardware de cifrado, realice los pasos que se describen en “Gestión de certificados en el hardware PKCS #11” en la página 573 para crear un repositorio de claves.

Emita el mandato siguiente para crear una solicitud de certificado con el mandato **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
```

donde:

-crypto *nombre_módulo*
Especifica el nombre completo de la biblioteca PKCS #11 proporcionada con el hardware de cifrado.

-tokenlabel *señal_hardware*
Especifica la etiqueta de señal del dispositivo criptográfico PKCS #11.

-pw *contraseña*
Especifica la contraseña para acceder al hardware de cifrado.

-label *etiqueta*
Especifica la etiqueta de certificado.

La etiqueta de un certificado TLS que utiliza IBM MQ es el valor del atributo **CERTLABL** si está establecido, o el valor predeterminado `ibmwebspheremq` con el nombre del gestor de colas o el ID de usuario IBM MQ MQI client añadido, todo en minúsculas. Para obtener más información, consulte "Etiquetas de certificados digitales, descripción de los requisitos" en la página 27.

-dn *nombre_distinguido*
Especifica el nombre distinguido X.500 especificado entre comillas dobles. Se necesita al menos un atributo en el nombre distinguido. Puede proporcionar varios atributos OU y DC.

Nota: El mandato **runmqakm** hace referencia al atributo de código postal como POSTALCODE, no PC. Especifique siempre POSTALCODE en el parámetro **-dn** cuando utilice el mandato **runmqakm** para solicitar certificados con un código postal.

-size *tamaño_clave*
Especifica el tamaño de clave. El valor puede ser 512, 1024o 2048.

-file *nombrearchivo*
Especifica el nombre de archivo para la solicitud de certificado.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que están validados por FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

-sig_alg
Especifica el algoritmo de hash que se utiliza cuando se crea la solicitud de certificado. Este algoritmo de hash se utiliza para crear la firma asociada con la solicitud de certificado. El valor puede 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` o `EC_ecdsa_with_SHA512`.

El valor predeterminado es `SHA1WithRSA`.

Para obtener más información sobre estos parámetros y los valores que se pueden especificar, consulte runmqakm -certreq.

Qué hacer a continuación

Envíe una solicitud de certificado a una CA. Cuando reciba el certificado firmado de la CA, añada el certificado firmado al repositorio de claves. Para obtener más información, consulte "Recepción de un certificado personal en el hardware PKCS #11" en la página 574.

ALW *Recepción de un certificado personal en el hardware PKCS #11*

Utilice este procedimiento para recibir un certificado personal para un gestor de colas o un IBM MQ MQI client en el hardware de cifrado.

Añada el certificado de CA de la CA que ha firmado el certificado personal al hardware criptográfico o al repositorio de claves secundario. Haga esto antes de recibir el certificado firmado en el hardware de cifrado. Para añadir un certificado de CA a un archivo de repositorio de claves, siga el procedimiento de "Adición de un certificado de CA, o la parte pública de un certificado de confianza, a un repositorio de claves en AIX, Linux, and Windows" en la página 560.

Emita el mandato siguiente para añadir un certificado personal a un repositorio de claves con el mandato **runmqakm** (GSKCapiCmd):

```
runmqakm -cert -receive -file filename -crypto module_name
         -tokenlabel hardware_token -pw hardware_password
         -format cert_format -fips
         -secondaryDB filename -secondaryDBpw password
```

donde:

-file *nombrearchivo*
Especifica el nombre de archivo completo del archivo que contiene el certificado personal.

-crypto *nombre_módulo*
Especifica el nombre completo de la biblioteca PKCS #11 proporcionada con el hardware de cifrado.

-tokenlabel *señal_hardware*
Especifica la etiqueta de señal del dispositivo criptográfico PKCS #11.

-pw *contraseña_hardware*
Especifica la contraseña para acceder al hardware de cifrado.

-format *formato_cert*
Especifica el formato del certificado. El valor puede ser `ascii` para datos ASCII codificados con Base64 o bien `binary` para datos DER binarios. El valor predeterminado es ASCII.

-fips
Especifica que el mandato se ejecuta en modalidad FIPS. Cuando está en modalidad FIPS, el componente IBM Crypto for C (ICC) utiliza algoritmos que están validados con FIPS 140-2. Si el componente ICC no se inicializa en modalidad FIPS, el mandato **runmqakm** falla.

-secondaryDB *nombrearchivo*
Especifica el nombre de archivo completo del archivo de repositorio de claves que se utiliza para almacenar el certificado de CA.

-secondaryDBpw *contraseña*
Especifica la contraseña del archivo de repositorio de claves que se utiliza para almacenar el certificado de CA.

Protección de contraseñas en archivos de configuración de componentes de IBM MQ

Para utilizar determinadas características de IBM MQ, es posible que tenga que proporcionar contraseñas utilizadas por la característica. Las contraseñas que se proporcionan a IBM MQ se pueden proteger utilizando un sistema de protección de contraseñas.

La lista siguiente explica la terminología que se utiliza para cada componente que procesa contraseñas cifradas:

Clave inicial
La clave de cifrado que se utiliza para proteger la contraseña.

Clave inicial predeterminada
La clave de cifrado predeterminada que se utiliza si no proporciona una clave inicial cuando la contraseña está cifrada.

Serie de texto sin formato
La serie que está cifrada, normalmente una contraseña.

Serie de contraseña cifrada
Una serie que contiene la contraseña cifrada en un formato que IBM MQ entiende.

Especificación de la clave inicial

Para cada componente, puede optar por especificar una clave inicial que se utiliza para cifrar contraseñas.

- Si no especifica una clave inicial, se utiliza la clave inicial predeterminada para el componente. La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Esto significa que una contraseña cifrada con la clave inicial predeterminada no está protegida de forma segura, ya que es posible que una instalación diferente pueda descifrar la contraseña.
- Si proporciona su propia clave inicial exclusiva, solo los usuarios con acceso a la clave inicial que proporcione pueden descifrar la contraseña.

Atención: Para proporcionar el nivel más alto de seguridad para las contraseñas almacenadas, proporcione una clave inicial exclusiva para cada componente de IBM MQ .

Si elige utilizar su propia clave inicial, especifique una clave inicial exclusiva para cada componente que se lista. La clave inicial se utiliza para proteger las contraseñas almacenadas en la configuración de dicho componente. La misma clave inicial también debe estar disponible para el componente para que se descifre la contraseña.

La mayoría de los componentes requieren que la clave inicial se proporcione en un archivo. La clave inicial contenida en el archivo de claves inicial debe cumplir los requisitos siguientes:

- Debe tener al menos un carácter de longitud.
- Debe ser una sola línea de texto.

La longitud máxima de la clave inicial es ilimitada y se puede especificar cualquier carácter. Para una seguridad adecuada, especifique una clave inicial que tenga al menos 16 caracteres de longitud. Por ejemplo, el archivo de claves inicial puede contener la siguiente serie:

```
Th1sIs@n3Ncypt|onK$y
```

El acceso al archivo de claves inicial debe estar limitado únicamente a los usuarios que necesitan acceder a la clave inicial utilizando los permisos de archivo del sistema operativo.

Para obtener más información sobre las ventajas y limitaciones de la protección por contraseña, consulte “Los límites de la protección a través del cifrado de contraseña” en la página 583.

Protección de contraseñas en cada componente de IBM MQ

Varios componentes de IBM MQ pueden proteger las contraseñas almacenadas. En función del componente, estas contraseñas se pueden proporcionar utilizando uno de los mecanismos siguientes:

- Se proporciona directamente al gestor de colas de IBM MQ o a IBM MQ client.
- Especificado en una variable de entorno.
- Se almacena en un archivo de configuración.

Cada componente proporciona un método para cifrar contraseñas. En la mayoría de los componentes, las contraseñas deben estar cifradas antes de que se suministren a IBM MQ o se almacenen en la configuración.

Importante: Una contraseña cifrada que se genera para su uso con un componente no se puede copiar en el archivo de configuración de otro componente. Una contraseña cifrada para que la utilice un componente determinado debe estar protegida con el programa de utilidad proporcionado por el mismo componente.

Los detalles sobre cómo proteger las contraseñas para cada componente de IBM MQ que da soporte a la protección por contraseña se listan en las secciones siguientes:

- Advanced Message Security
- “Managed File Transfer” en la página 578
- “IBM MQ Internet Pass-Thru” en la página 579
- “IBM MQ clients que utilizan hardware criptográfico” en la página 579
- “Gestor de colas IBM MQ” en la página 580
- “Aplicaciones cliente C de IBM MQ” en la página 581
- V 9.4.0 “Configuraciones de HA nativa” en la página 581
- V 9.4.0 “Gestor de colas de IBM MQ (stanzaAuthToken en el archivo qm.ini)” en la página 582

Advanced Message Security

Los clientes Advanced Message Security (AMS) Java requieren acceso a un almacén de claves que contiene las claves privadas que se utilizan para proteger los mensajes.

Advanced Message Security (AMS) Los clientes MQI o gestores de colas que están configurados para realizar la intercepción de MCA pueden requerir acceso al hardware criptográfico PKCS#11 o a archivos PEM que contienen las claves privadas que se utilizan para proteger los mensajes.

Para acceder a estos repositorios de claves, se debe proporcionar una contraseña en el archivo de configuración AMS denominado `keystore.conf`. Utilice el mandato **runamscred** para proteger la información confidencial contenida en el archivo `keystore.conf` . Por ejemplo,

```
runamscred -f <keystore configuration file>
```

El mandato **runamscred** protege los parámetros confidenciales dentro del archivo que se especifica utilizando el parámetro **-f** .

Hay dos mandatos **runamscred** disponibles en una instalación de IBM MQ :

- Un mandato MQI **runamscred** que se encuentra en `<IBM MQ installation root>/bin`
- Un mandato de Java **runamscred** que se encuentra en `<IBM MQ installation root>/java/bin`

Atención: Para garantizar la compatibilidad,

1. Utilice el mandato Java **runamscred** para proteger los archivos de configuración que se utilizan con los clientes de Java AMS y el mandato MQI **runamscred** para proteger los archivos de configuración para IBM MQ MQI clients que utilizan AMS.
2. Verifique que toda la información confidencial necesaria está protegida después de ejecutar el mandato **runamscred** .
3. Proporcione el archivo que contiene la contraseña protegida de forma normal para las aplicaciones habilitadas para AMS .

De forma predeterminada, el mandato **runamscred** cifra la contraseña en el archivo de configuración con la clave inicial predeterminada. Para cifrar las contraseñas con una clave inicial específica, utilice uno de los mecanismos siguientes para especificar el nombre del archivo que contiene la clave inicial, en orden de prioridad:

1. El parámetro **-sf** para el mandato **runamscred** .
2. La variable de entorno **MQS_AMSCRED_KEYFILE** .
3. El parámetro **amscred.keyfile** en el archivo de configuración `keystore.conf` .

PRECAUCIÓN: La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Para proteger las contraseñas de forma segura, proporcione una clave inicial que sea exclusiva para la instalación cuando cifre las contraseñas.

Si especifica un archivo de claves inicial al ejecutar el mandato **runamscred** para cifrar las contraseñas en la configuración de AMS , también debe especificar el mismo archivo de claves inicial cuando se ejecuten las aplicaciones AMS . Se pueden utilizar los mecanismos siguientes para especificar el nombre del archivo de claves inicial, en orden de prioridad:

1. La variable de entorno **MQS_AMSCRED_KEYFILE** .
2. El parámetro **amscred.keyfile** en el archivo de configuración `keystore.conf` .

De forma predeterminada, el mandato **runamscred** protege las credenciales con un sistema de protección que no es compatible con versiones de AMS anteriores a IBM MQ 9.2. Para proteger los archivos de configuración con el sistema de protección de credenciales que es compatible con versiones anteriores a IBM MQ 9.2, especifique el parámetro **-sp 0** cuando se ejecute el mandato **runamscred** .

Managed File Transfer

Managed File Transfer (MFT) almacena las credenciales necesarias para acceder a los gestores de colas y a otros recursos en los siguientes archivos de propiedades XML:

MQMFTCredentials.xml
Este archivo contiene las credenciales siguientes:

- Credenciales que se utilizan para conectarse a gestores de colas de agente, coordinación y mandatos.
- Contraseñas que se utilizan para acceder a los almacenes de claves que se utilizan para las comunicaciones seguras.

ProtocolBridgeCredentials.xml
Este archivo contiene credenciales que se utilizan para conectarse a servidores de protocolo, como FTP, SFTP y FTPS.

ConnectDirectCredentials.xml
Este archivo contiene las credenciales que utiliza un agente de Connect:Direct para conectarse a un nodo Connect:Direct .

Para proteger la información confidencial que se almacena en estos archivos, utilice el mandato fteObfuscate . Especifique el nombre del archivo que se va a proteger utilizando el distintivo **-f** . Por ejemplo:

```
fteObfuscate -f <File to protect>
```

De forma predeterminada, el mandato **fteObfuscate** protege las credenciales con la clave inicial predeterminada. Para proteger las credenciales con una clave inicial específica, utilice el parámetro **-sf** para especificar la vía de acceso al archivo que contiene la clave inicial. Por ejemplo:

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

PRECAUCIÓN: La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Para proteger las contraseñas de forma segura, proporcione una clave inicial que sea exclusiva para la instalación cuando cifre las contraseñas.

Atención:

1. Verifique que toda la información confidencial está protegida después de ejecutar **fteObfuscate**.
2. Proporcione el archivo protegido como normal a MFT.

Si especifica un archivo de claves inicial al ejecutar el mandato **fteObfuscate** para proteger las credenciales en la configuración de MFT , también debe especificar el mismo archivo de claves inicial cuando se inicie MFT . Se pueden utilizar los mecanismos siguientes para especificar el nombre del archivo de claves inicial, en orden de prioridad:

1. La propiedad del sistema **com.ibm.wmqfte.cred.keyfile** Java .

   Nota: Antes de IBM MQ 9.3.1 y IBM MQ 9.3.0 Fix Pack 10, el nombre de esta propiedad del sistema Java se ha escrito incorrectamente como **com.ibm.wqmfte.cred.keyfile**. A partir de IBM MQ 9.3.1 y IBM MQ 9.3.0 Fix Pack 10, Managed File Transfer utiliza ambas versiones de la propiedad del sistema Java para mantener la compatibilidad con versiones anteriores. Si se establecen ambas propiedades del sistema Java , se utiliza el valor de la propiedad **com.ibm.wmqfte.cred.keyfile** correctamente escrita.
2. Propiedades del agente, registrador, mandatos y archivos de propiedades de coordinación.
3. La propiedad **commonCredentialsKeyFile** en el archivo `installation.properties` .

Para obtener más información, consulte “Cifrado de credenciales almacenadas en MFT” en la página 585.

De forma predeterminada, el mandato **fteObfuscate** protege las credenciales con un sistema de protección que no es compatible con versiones de MFT anteriores a IBM MQ 9.2. Para proteger los archivos de configuración con el sistema de protección de credenciales que es compatible con versiones anteriores a IBM MQ 9.2, especifique el parámetro **-sp 0** cuando se ejecute el mandato **fteObfuscate** .

IBM MQ Internet Pass-Thru

El archivo de configuración IBM MQ Internet Pass-Thru (MQIPT) puede contener contraseñas que se utilizan para acceder a diversos recursos.

Proteja las contraseñas en el archivo de configuración MQIPT utilizando el mandato **mqiptPW** .

El mandato **mqiptPW** solicita que se especifique la contraseña que se va a cifrar y devuelve la contraseña cifrada. Copie la contraseña cifrada en el archivo de configuración MQIPT .

De forma predeterminada, el mandato **mqiptPW** cifra una contraseña con la clave inicial predeterminada. Para cifrar la contraseña con una clave inicial específica, utilice el parámetro **-sf** para especificar la vía de acceso al archivo que contiene la clave inicial.

PRECAUCIÓN: La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Para proteger las contraseñas de forma segura, proporcione una clave inicial que sea exclusiva para la instalación cuando cifre las contraseñas.

Para obtener más información, consulte Especificación de la clave de cifrado de contraseña.

Si especifica un archivo de claves inicial al cifrar la contraseña del repositorio de claves, también debe especificar el mismo archivo de claves inicial cuando se inicie MQIPT . Se pueden utilizar los mecanismos siguientes para especificar el nombre del archivo de claves inicial, en orden de prioridad:

1. El parámetro **-sf** en el mandato que se utiliza para iniciar MQIPT.
2. la variable de entorno **MQS_MQIPTCRED_KEYFILE**.
3. la propiedad **com.ibm.mq.ipt.cred.keyfile** Java.
4. Un archivo denominado `mqipt_cred.key` en el directorio de inicio de MQIPT . El directorio de inicio de MQIPT es el directorio que contiene el archivo de configuración MQIPT .

De forma predeterminada, el mandato **mqiptPW** protege las credenciales con un sistema de protección que no es compatible con versiones de MQIPT anteriores a IBM MQ 9.2. Para proteger las contraseñas con el sistema de protección de credenciales que es compatible con versiones anteriores a IBM MQ 9.2, utilice la sintaxis del mandato **mqiptPW** que está soportada en versiones anteriores a IBM MQ 9.2.

IBM MQ clients que utilizan hardware criptográfico

Puede configurar clientes IBM MQ para que utilicen hardware criptográfico PKCS #11 para almacenar claves privadas y certificados que se utilizan en las comunicaciones TLS. Para acceder a los dispositivos PKCS #11 , debe proporcionar una contraseña como parte de la serie de configuración que se proporciona a IBM MQ client.

Importante: Las contraseñas proporcionadas utilizando el campo **CryptoHardware** en la estructura MQSCO, o el atributo **SSLCRYP** del gestor de colas no se pueden proteger utilizando este mecanismo.

Puede proteger esta contraseña utilizando el mandato **runp11cred** , que se puede encontrar en la carpeta `bin` del directorio de instalación de IBM MQ .

El mandato **runp11cred** solicita que se especifique la contraseña que se va a cifrar y devuelve la contraseña cifrada. La contraseña cifrada debe copiarse en la serie de configuración de hardware criptográfico.

Por ejemplo, si la serie de configuración de hardware criptográfico es la siguiente:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;Passw0rd;SYMMETRIC_CIPHER_ON
```

Cuando el mandato **runp11cred** le solicite que especifique la contraseña, especifique Passw0rd. El mandato devuelve una serie que es similar a la siguiente:

```
<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==
```

Sustituya la contraseña en la serie de configuración de hardware criptográfico por la serie devuelta por el mandato **runp11cred** , para proporcionar la serie siguiente que contiene la contraseña cifrada:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHm SNF0/
ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

Cuando se ejecute la aplicación IBM MQ client , especifique la serie de configuración de hardware criptográfico que contiene la contraseña cifrada en uno de los métodos siguientes:

- El atributo **SSLCryptoHardware** en la stanza SSL del archivo de configuración del cliente.
- La variable de entorno **MQSSLCRYP** .

De forma predeterminada, el mandato **runp11cred** cifra una contraseña con una clave inicial predeterminada. Para proteger una contraseña con su propia clave inicial, especifique el nombre del archivo que contiene la clave inicial utilizando uno de los mecanismos siguientes, en orden de prioridad:

1. El parámetro **-sf** para el mandato **runp11cred** .
2. la variable de entorno **MQS_SSLCRYP_KEYFILE**.

PRECAUCIÓN: La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Para proteger las contraseñas de forma segura, proporcione una clave inicial que sea exclusiva para la instalación cuando cifre las contraseñas.

Si especifica un archivo de claves inicial al cifrar la contraseña del repositorio de claves, también debe especificar el nombre del archivo que contiene la clave inicial cuando se ejecuta IBM MQ client . Especifique el nombre del archivo de claves inicial utilizando uno de los mecanismos siguientes, en orden de prioridad:

1. La variable de entorno **MQS_SSLCRYP_KEYFILE** .
2. El atributo **SSLCryptoHardwareKeyFile** en la stanza **SSL** del archivo de configuración del cliente.

Gestor de colas IBM MQ

El gestor de colas de IBM MQ almacena las contraseñas internamente en varios atributos. Por ejemplo, el atributo **KEYRPWD** del gestor de colas. El gestor de colas cifra automáticamente la contraseña antes de que se almacene en archivos en disco.

La contraseña del repositorio de claves TLS del gestor de colas se puede proteger utilizando el sistema de protección de contraseñas de IBM MQ o un archivo de ocultación de repositorio de claves. Para obtener más información sobre estos dos métodos, consulte “Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows” en la página 302.

Cuando el gestor de colas cifra una contraseña, se utiliza la clave inicial predeterminada a menos que especifique su propia clave inicial. Para utilizar su propia clave inicial, establezca el atributo **INITKEY** del gestor de colas en una clave exclusiva y fuerte antes de establecer cualquier atributo de gestor de colas cifrado.

PRECAUCIÓN: La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Para proteger las contraseñas de forma segura, proporcione una clave inicial que sea exclusiva para la instalación cuando cifre las contraseñas.

Aviso: Si la clave inicial se modifica después de establecer el valor de los atributos cifrados, los atributos cifrados no se vuelven a cifrar con la nueva clave inicial. Por lo tanto, al cambiar la clave inicial sin volver a proporcionar la frase de contraseña del repositorio de claves, IBM MQ no puede descifrar la frase de contraseña del repositorio de claves y no puede acceder al repositorio de claves.

Para obtener más información, consulte INITKEY.

Aplicaciones cliente C de IBM MQ

Las bibliotecas de cliente C de IBM MQ requieren contraseñas para acceder a determinados recursos protegidos. Por ejemplo, un repositorio de claves TLS para aplicaciones que utilizan TLS para conectarse al gestor de colas.

La contraseña del repositorio de claves se puede proteger utilizando el sistema de protección de contraseñas de IBM MQ o un archivo de ocultación de repositorio de claves. Para obtener más información sobre estos dos métodos, consulte "Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows" en la página 302.

Para proteger las contraseñas con el sistema de protección de contraseñas de IBM MQ , utilice el mandato **runmqicred** . El mandato se encuentra en el directorio *MQ_INSTALLATION_PATH*/bin .

El mandato **runmqicred** solicita que se especifique la contraseña que se va a cifrar y devuelve la contraseña cifrada. La aplicación cliente puede utilizar la contraseña cifrada en lugar de una contraseña de texto sin formato.

Por ejemplo, si elige proporcionar una contraseña de repositorio de claves TLS utilizando la variable de entorno *MQKEYRPWD* y la contraseña de almacén de claves TLS es Passw0rd. Cuando ejecute **runmqicred**, especifique Passw0rd cuando se le solicite. El mandato devuelve una serie que es similar a la siguiente:

```
<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w==
```

Establezca esta serie como valor para la variable de entorno *MQKEYRPWD* :

```
export MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
set MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
```

De forma predeterminada, el mandato **runmqicred** cifra una contraseña con la clave inicial predeterminada. Para proteger una contraseña con su propia clave inicial, utilice uno de los mecanismos siguientes para especificar el nombre del archivo que contiene la clave, en orden de prioridad:

1. El parámetro **-sf** para el mandato **runmqicred** .
2. La variable de entorno *MQS_MQI_KEYFILE* .

PRECAUCIÓN: La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Para proteger las contraseñas de forma segura, proporcione una clave inicial que sea exclusiva para la instalación cuando cifre las contraseñas.

Si especifica un archivo de claves inicial al cifrar la contraseña, también debe hacer que la clave inicial esté disponible para la aplicación cliente cuando se ejecute.

Para obtener más información, consulte "Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows" en la página 308.

Configuraciones de HA nativa

V 9.4.0

El tráfico de réplica de registro HA nativo entre instancias se puede cifrar utilizando TLS. Los certificados que se utilizan para proteger el tráfico de réplica de registro se almacenan en un repositorio de claves que se especifica en la stanza **NativeHALocalInstance** del archivo qm.ini .

La contraseña del repositorio de claves se puede proteger utilizando el sistema de protección de contraseñas de IBM MQ o un archivo de ocultación de repositorio de claves. Para obtener más información sobre estos dos métodos, consulte "Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows" en la página 302.

Para proteger la contraseña del repositorio de claves de HA nativa con el sistema de protección de contraseñas de IBM MQ , utilice el mandato **runmqicred** .

El mandato **runmqicred** solicita que se especifique la contraseña que se va a cifrar y devuelve la contraseña cifrada. Se debe utilizar la contraseña cifrada en lugar de una contraseña de texto sin formato.

Establezca el valor del atributo **KeyRepositoryPassword** en la stanza **NativeHALocalInstance** del archivo `qm.ini` en la contraseña cifrada que devuelve el mandato.

De forma predeterminada, el mandato **runmqicred** cifra una contraseña con la clave inicial predeterminada. Para proteger una contraseña con su propia clave inicial, utilice uno de los mecanismos siguientes para especificar el nombre del archivo que contiene la clave, en orden de prioridad:

1. El parámetro **-sf** para el mandato **runmqicred** .
2. La variable de entorno *MQS_MQI_KEYFILE* .

PRECAUCIÓN: La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Para proteger las contraseñas de forma segura, proporcione una clave inicial que sea exclusiva para la instalación cuando cifre las contraseñas.

Si especifica un archivo de claves inicial al cifrar la contraseña del repositorio de claves, también debe especificar el mismo archivo de claves inicial utilizando el atributo **InitialKeyFile** en la stanza **NativeHALocalInstance** del archivo `qm.ini` .

Para obtener más información, consulte NativeHALocalStanza de instancia del archivo qm.ini.

Gestor de colas de IBM MQ (stanzaAuthToken en el archivo `qm.ini`)

A partir de IBM MQ 9.3.4, IBM MQ MQI clients que se conectan a gestores de colas de IBM MQ que se ejecutan en sistemas AIX o Linux , pueden utilizar señales de autenticación para autenticarse con el gestor de colas. El gestor de colas debe estar configurado para aceptar señales de autenticación y poder acceder al certificado de clave pública del emisor de señales o a la clave secreta que se utiliza para firmar la señal. El repositorio de claves que contiene los certificados de clave pública o claves secretas del emisor de confianza está protegido con una contraseña.

La contraseña del repositorio de claves se puede proteger utilizando el sistema de protección de contraseñas de IBM MQ o un archivo de ocultación de repositorio de claves. Para obtener más información sobre estos dos métodos, consulte “Cifrado de contraseñas de repositorio de claves en AIX, Linux, and Windows” en la página 302.

Para proteger la contraseña del repositorio de claves de señal de autenticación con el sistema de protección de contraseñas de IBM MQ , utilice el mandato **runqmcred** para cifrar la contraseña.

El mandato **runqmcred** solicita que se especifique la contraseña que se va a cifrar y devuelve la contraseña cifrada. Se debe utilizar la contraseña cifrada en lugar de una contraseña de texto sin formato. Copie la contraseña cifrada en un archivo e incluya la vía de acceso al archivo en el atributo **KeyStorePwdFile** de la stanza **AuthToken** en el archivo `qm.ini` .

De forma predeterminada, el mandato **runqmcred** cifra una contraseña con la clave inicial predeterminada. Para cifrar la contraseña con una clave inicial específica, utilice el parámetro **-sf** para especificar la vía de acceso al archivo que contiene la clave inicial.

PRECAUCIÓN: La clave inicial predeterminada es la misma para todas las instalaciones de IBM MQ . Para proteger las contraseñas de forma segura, proporcione una clave inicial que sea exclusiva para la instalación cuando cifre las contraseñas.

Importante: Si proporciona una clave inicial al cifrar la contraseña, se debe especificar la misma clave inicial en el atributo **INITKEY** del gestor de colas para que el gestor de colas pueda descifrar la contraseña. Si el atributo **INITKEY** del gestor de colas ya está establecido, utilice la misma clave inicial cuando ejecute el mandato **runqmcred** . Para obtener más información sobre el atributo **INITKEY** del gestor de colas, consulte INITKEY.

Por ejemplo, para cifrar las contraseñas de almacén de claves de señal de autenticación con la clave inicial en el archivo `/home/initial.key`, emita el mandato siguiente:

```
runqmcred -sf /home/initial.key
```

Para obtener más información, consulte “Configuración de un gestor de colas para aceptar señales de autenticación utilizando un almacén de claves local” en la página 338.

Los límites de la protección a través del cifrado de contraseña

IBM MQ da soporte al cifrado AES-128 para contraseñas almacenadas en varios archivos de configuración. Cuando se utiliza el cifrado AES (Advanced Encryption Standard) para proteger las contraseñas en la configuración de IBM MQ , es necesario comprender los límites de la protección que proporciona.

El cifrado de una contraseña en los archivos de configuración de IBM MQ no significa que la contraseña esté protegida o protegida. Solo evita que la contraseña sea fácilmente recuperable por alguien que pueda acceder a la contraseña cifrada, pero no conoce la clave de cifrado. Los procesos de IBM MQ requieren acceso a la contraseña cifrada y a la clave de descifrado para obtener la contraseña de texto simple para utilizarla. Ambos elementos de datos deben almacenarse en el sistema de archivos en una ubicación accesible para IBM MQ. Cualquier persona que cifre una contraseña que se coloca en un archivo de configuración también requiere acceso a la clave de cifrado. Si un atacante tiene acceso al mismo conjunto de archivos que IBM MQ, la aplicación del cifrado AES a la contraseña sólo proporciona un nivel mínimo de protección.

No obstante, el cifrado de contraseñas en reposo es importante considerarlo ya que evita la divulgación accidental de contraseñas y permite compartir los archivos de configuración, si la clave de descifrado no se comparte también.

Además de asegurarse de que el archivo que contiene la clave de descifrado no se comparte, debe tener cuidado de asegurarse de que el archivo está protegido de otros usuarios del sistema. Aunque los archivos de configuración de IBM MQ pueden ser accesibles para todos los usuarios, restrinja los permisos del archivo que contiene la clave de descifrado al mínimo necesario. A los ID de usuario con los que se ejecutan los procesos de IBM MQ se les debe otorgar acceso para leer el archivo que contiene la clave de descifrado. Sin embargo, no es necesario otorgar acceso para leer el archivo a un grupo o a todos los usuarios del sistema.

Protección de los detalles de autenticación de base de datos

Si está utilizando la autenticación de nombre de usuario y contraseña para conectarse al gestor de base de datos, puede almacenarlos en el almacén de credenciales XA de MQ para evitar almacenar la contraseña en texto sin formato en el archivo `qm.ini`.

Actualizar XAOpenString para el gestor de recursos

Para utilizar el almacén de credenciales debe modificar XAOpenString en el archivo `qm.ini`. La serie se utiliza para conectarse al gestor de base de datos. Especifique campos sustituibles para identificar donde el nombre de usuario y la contraseña se sustituyen dentro de la serie XAOpenString.

- El campo +USER+ se sustituye por el valor de nombre de usuario almacenado en el almacén XACredentials.
- El campo +PASSWORD+ se sustituye por el valor de contraseña almacenado en el almacén XACredentials.

Los siguientes ejemplos muestran cómo modificar una XAOpenString para que utilice el archivo de credenciales para conectarse a la base de datos.

Conexión con una base de datos Db2

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

Conexión a una base de datos Oracle

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

Trabajar con las credenciales para la base de datos para el almacén de credenciales XA de MQ

Después de actualizar el archivo `qm.ini` con las series de credenciales sustituibles, deberá añadir el nombre de usuario y la contraseña al almacén de credenciales de MQ mediante el mandato **setmqxacred**. También puede utilizar **setmqxacred** para modificar credenciales existentes, suprimir credenciales o listar credenciales. Los siguientes ejemplos proporcionan algunos casos de uso típicos:

Adición de credenciales

El siguiente mandato guarda de forma segura el nombre de usuario y la contraseña para el gestor de colas QM1 para el recurso mqdb2.

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

Actualización de credenciales

Para actualizar el nombre de usuario y la contraseña que se utilizan para conectarse a una base de datos, vuelva a emitir el mandato **setmqxacred** con el nombre de usuario y la contraseña nuevos.

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

Debe reiniciar el gestor de colas para que los cambios entren en vigor.

Supresión de credenciales

El siguiente mandato suprime las credenciales:

```
setmqxacred -m QM1 -x mydb2 -d
```

Listado de credenciales

El siguiente mandato lista credenciales:

```
setmqxacred -m QM1 -l
```

Referencia relacionada

setmqxacred

Protección de Managed File Transfer

Directamente tras la instalación y sin ninguna modificación, Managed File Transfer tiene un nivel de seguridad que puede ser adecuado para realizar pruebas o evaluaciones en un entorno protegido. Sin embargo, en un entorno de producción, debe considerar la posibilidad de controlar de manera apropiada quién puede iniciar operaciones de transferencia de archivos, quién puede leer y grabar los archivos que se están transfiriendo y cómo proteger la integridad de los archivos.

Tareas relacionadas

Gestión de autorizaciones de grupo para recursos específicos de MFT

Gestión de autorizaciones para recursos específicos de MFT

“Utilización de Advanced Message Security con Managed File Transfer” en la página 651

Este caso de ejemplo explica cómo configurar Advanced Message Security para proporcionar privacidad de mensajes para los datos que se envían a través de Managed File Transfer.

Referencia relacionada

Autorizaciones para el acceso de MFT a los sistemas de archivos

Propiedad commandPath de MFT

Autorización para publicar mensajes de registro y estado de agentes MFT

Cifrado de credenciales almacenadas en MFT

Managed File Transfer (MFT) requiere varios ID de usuario y credenciales, que se almacenan en dos archivos XML, y puede enmascararlos utilizando el mandato **fteObfuscate** .

Archivos de credenciales

MQMFTCredentials.xml
Este archivo contiene el ID de usuario y las credenciales para conectarse a agentes y gestores de colas de coordinación y mandatos. Las credenciales para acceder a almacenes de claves para conexiones seguras con gestores de colas también se almacenan en el mismo archivo.

Consulte “Autenticación de conexión de MFT y IBM MQ” en la página 588 para obtener detalles de los valores de propiedad que definen la ubicación del archivo `MQMFTCredentials.xml` .

ProtocolBridgeCredentials.xml
Este archivo contiene el ID de usuario y las credenciales para conectarse a los servidores de protocolo.

Cifrado de credenciales utilizando el mandato `fteObfuscate`

El mandato **fteObfuscate** acepta los parámetros siguientes:

- **-f** *nombre_archivo_credenciales* (necesario)

  Nota: Deprecated Este parámetro sustituye al parámetro **-credentialsFile** que está en desuso desde IBM MQ 9.2.0.
- **-sp** *modalidad_protección*
- **-sf** *archivo_claves_credenciales*
- **-o** *nombre_de_archivo_de_salida*

Consulte **fteObfuscate** para obtener detalles de los parámetros.

Si no especifica la modalidad de protección, o un archivo de claves de credenciales, el mandato utiliza la modalidad de protección predeterminada y utiliza el algoritmo más reciente, pero con una clave fija para cifrar las credenciales.

Si especifica una modalidad de protección de `0`y no especifica un archivo de claves de credenciales, el mandato funciona como en los releases anteriores del producto. Recibe un mensaje de aviso en la consola que indica el uso de la protección en desuso.

Si especifica una modalidad de protección de `0`y especifica un archivo de claves de credenciales, recibirá una salida de error en la consola que indica que no es válido especificar el archivo de claves cuando se utiliza la modalidad de protección `0`.

Si especifica la modalidad de protección de `1`y no especifica un archivo de claves de credenciales, el mandato utiliza el algoritmo más reciente, pero con una clave fija para cifrar las credenciales.

Si especifica la modalidad de protección de `1`y especifica un archivo de claves de credenciales, el mandato cifra las credenciales con el algoritmo más reciente.

Si especifica la modalidad de protección de `1`, o no especifica la modalidad de protección, y especifica un archivo de claves de credenciales que no existe, se genera un error en la consola que indica que el archivo no existe.

Si especifica la modalidad de protección de 1, o no especifica la modalidad de protección, y especifica un archivo de claves de credenciales que no es legible, se genera un error en la consola que indica que el archivo no es legible.

Si especifica la modalidad de protección de 2y no especifica un archivo de claves de credenciales, el mandato utiliza la modalidad de protección 2 para cifrar las credenciales utilizando el algoritmo más reciente y una clave fija para cifrar.

Si especifica la modalidad de protección de 2y especifica un archivo de claves de credenciales, el mandato utiliza la modalidad de protección 2 para cifrar las credenciales utilizando el algoritmo más reciente y una clave especificada por el usuario para cifrar.

Si especifica la modalidad de protección de 2, o no especifica la modalidad de protección, y especifica un archivo de claves de credenciales que no existe, se genera un error en la consola que indica que el archivo no existe.

Si especifica la modalidad de protección de 2, o no especifica la modalidad de protección, y especifica un archivo de claves de credenciales que no es legible, se genera un error en la consola que indica que el archivo no es legible.

Descifrado de credenciales

Puede especificar la vía de acceso al archivo de claves inicial en varios lugares. Para descifrar las credenciales que se han cifrado utilizando una clave inicial distinta de la predeterminada, el nombre del archivo que contiene la clave inicial debe proporcionarse a MFT de una de las maneras siguientes, en este orden de prioridad:

1. Utilizando una propiedad del sistema Java , por ejemplo:

   ```
   -Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
   ```

   Nota:

   - Antes de IBM MQ 9.3.1 y IBM MQ 9.3.0 Fix Pack 10, el nombre de esta propiedad del sistema Java se ha escrito incorrectamente en el código de producto como `com.ibm.wqmfte.cred.keyfile`. A partir de IBM MQ 9.3.1 y IBM MQ 9.3.0 Fix Pack 10, la ortografía del nombre de propiedad se corrige como `com.ibm.wmqfte.cred.keyfile`. Managed File Transfer utiliza ambas versiones de la propiedad del sistema Java al comprobar si un usuario ha especificado un archivo que contiene la clave inicial que se debe utilizar para cifrar y descifrar credenciales. Esto le permite utilizar la ortografía correcta del nombre de propiedad, manteniendo al mismo tiempo la compatibilidad con el antiguo nombre mal escrito. Tenga en cuenta que si se establecen ambas propiedades del sistema Java , se utiliza el valor de la propiedad `com.ibm.wmqfte.cred.keyfile` correctamente escrita.
   - Antes de IBM MQ 9.3.1 y IBM MQ 9.3.0 Fix Pack 10, utilice la propiedad `com.ibm.wqmfte.cred.keyfile`.

   .
2. Estableciendo una propiedad en un archivo de propiedades de agente, mandato, coordinación o registrador. El nombre del archivo de propiedades y la propiedad que se debe establecer en él se muestran en la tabla siguiente:

Archivo de propiedades	Nombre de propiedad
agent.properties	agentCredentialsKeyFile
command.properties	commandCredentialsKeyFile
coordination.properties	coordinationCredentialsKeyFile
logger.properties	loggerCredentialsKeyFile
3. En el archivo installation.properties .

En lugar de añadir propiedades en archivos de propiedades individuales, puede añadir la propiedad **commonCredentialsKeyFile** al archivo `installation.properties` común existente, para que el agente, el registrador y los mandatos puedan utilizar la misma propiedad.

Si ha definido las diversas propiedades de **CredentialsKeyFile** en varias ubicaciones:

- La vía de acceso del archivo de claves de credenciales que se utiliza para el agente y el registrador se registra en el archivo `output0.log` para dicho agente o registrador.
- La vía de acceso del archivo de claves de credenciales que se utiliza para los mandatos se visualiza en la consola.

La Java propiedad del sistema **com.ibm.wmqfte.cred.keyfile** altera temporalmente todas las demás. Si la propiedad del sistema no está establecida, el agente busca en el archivo `agent.properties` , seguido del archivo `installation.properties` para el archivo de claves inicial.

Si el archivo de claves inicial sigue sin encontrarse y ha establecido la modalidad de protección en el mandato **fteObfuscate** en 1, el agente registra un mensaje de error en el archivo `output0.log` .

Si ha establecido la modalidad de protección en `0` en el mandato **fteObfuscate** , se registra un mensaje de aviso que indica que está en desuso.

El registrador y los mandatos siguen los mismos pasos para localizar el archivo de claves inicial.

Puente de protocolo y Puente de Connect:Direct

Protocol Bridge utiliza un archivo de propiedades, `ProtocolBridgeProperties.xml`, para conectarse a los servidores FTP, SFTP y FTPS. Este archivo de propiedades contiene los atributos de conexión necesarios para conectarse a estos servidores.

Es necesario reiniciar el agente de puente si modifica el valor de los atributos **credentialsFile** o **credentialsKeyFile** en el archivo `ProtocolBridgeProperties.xml` .

Uno de los atributos es **credentialsFile**, y el valor contiene la vía de acceso a un archivo XML que contiene UID, o PWD, o Clave necesaria para conectarse a estos servidores. El valor predeterminado para el atributo es *ProtocolBridgeCredentials.xml* y el archivo está en el directorio de inicio, al igual que el archivo `MQMFTCredentials.xml` .

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

Al igual que *MQMFTCredentails.xml*, puede cifrar *ProtocolBridgeCredentials.xml* con el mandato **fteObfuscate** . Para fines de descifrado, puede especificar la vía de acceso necesaria a un archivo de claves de credenciales utilizando el elemento adicional **credentialsKeyFile** tal como se muestra en el texto siguiente. La vía de acceso puede contener variables de entorno.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Nota: La especificación de un valor para la propiedad de agente **agentCredentialsKeyFile** , la propiedad **commonCredentialsKeyFile** en `installation.propertieso` a través de la propiedad del sistema **com.ibm.wqmfte.cred.keyfile**, no tiene ningún impacto en el valor especificado para el atributo **credentialsKeyFile** .

De forma similar, Connect:Direct Bridge utiliza *ConnectDirectNodeProperties.xml* para conectarse al servidor de Connect:Direct . El archivo XML contiene la información de conexión necesaria, junto con un atributo que define la vía de acceso al archivo XML de credenciales. Este archivo XML de credenciales contiene UID, o PWD, e información adicional necesaria para conectarse al servidor de Connect:Direct .

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

Al igual que el archivo *ProtocolBridgeCredentials.xml* , puede cifrar *ConnectDirectCredentials.xml* con el mandato **fteObfuscate** . Para fines de descifrado, puede especificar la vía de acceso necesaria a un

archivo de claves de credenciales utilizando el elemento adicional **credentialsKeyFile** tal como se muestra en el texto siguiente. La vía de acceso puede contener variables de entorno.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Nota: La especificación de un valor para la propiedad de agente **agentCredentialsKeyFile** , la propiedad **commonCredentialsKeyFile** en `installation.propertieso` a través de la propiedad del sistema **com.ibm.wqmfte.cred.keyfile** no tiene ningún impacto en el valor especificado para el atributo **credentialsKeyFile** .

Puede especificar el elemento **credentialsKeyFile** , sin especificar el elemento **credentialsFile** en el archivo *ProtocolBridgeProperties.xml* .

Si no especifica el elemento **credentialsFile** , el agente de puente de protocolo utiliza el archivo de credenciales predeterminado *ProtocolBridgeCredentials.xml* y el valor del archivo de claves especificado en el atributo **credentialsKeyFile** se utiliza para descifrar el archivo de credenciales.

De forma similar, puede especificar el elemento **credentialsKeyFile** , sin especificar el elemento **credentialsFile** en el archivo *ConnectDirectNodeProperties.xml* .

Si no especifica el elemento **credentialsFile** , el puente Connect:Direct utiliza el archivo de credenciales predeterminado *ConnectDirectCredentials.xml* y el valor del archivo de claves especificado en el atributo **credentialsKeyFile** se utiliza para descifrar el archivo de credenciales.

Utilización de la clave del conjunto de datos en z/OS

z/OS

En z/OS, puede especificar **MQMFTCredentials** y proporcionar el archivo de claves de credenciales utilizando un PDSE. Consulte “Configuring MQMFTCredentials.xml on z/OS” en la página 591

Referencia relacionada

Qué mandato de MFT se conecta a qué gestor de colas

Formato del archivo de credenciales de MFT

fteObfuscate (cifrar datos confidenciales)

Autenticación de conexión de MFT y IBM MQ

La autenticación de conexión permite que un gestor de colas se configure para autenticar aplicaciones utilizando un ID de usuario y una contraseña proporcionados. Si el gestor de colas asociado tiene la seguridad habilitada y requiere detalles de credenciales (ID de usuario y contraseña), la característica de autenticación de conexión debe estar habilitada antes de que se pueda realizar una conexión correcta a un gestor de colas. La autenticación de conexión se puede ejecutar en modalidad de compatibilidad o en modalidad de autenticación MQCSP.

Métodos para suministrar detalles de credenciales

Muchos mandatos de Managed File Transfer dan soporte a los métodos siguientes para suministrar detalles de credenciales:

Detalles proporcionados por los argumentos de línea de mandatos.
Los detalles de las credenciales se pueden especificar utilizando los parámetros **-mquserid** y **-mqpassword**. Si no se proporciona **-mqpassword**, se solicita al usuario la contraseña en la que no se visualiza la entrada.

Los detalles proporcionados desde un archivo de credenciales: `MQMFTCredentials.xml`.
Los detalles de credenciales pueden predefinirse en un archivo `MQMFTCredentials.xml` como texto simple o texto enmascarado.

Multi Para obtener información sobre cómo configurar un archivo MQMFTCredentials.xml en IBM MQ for Multiplatforms , consulte “Configuración de MQMFTCredentials.xml en Multiplatforms” en la página 589.

z/OS Para obtener información sobre cómo configurar un archivo MQMFTCredentials.xml en IBM MQ for z/OS , consulte “Configuring MQMFTCredentials.xml on z/OS” en la página 591.

Prioridad

La prioridad de determinar los detalles de credenciales es:

1. Argumento de línea de mandatos.
2. Índice de `MQMFTCredentials.xml` por gestor de colas asociado y usuario que ejecuta el mandato.
3. Índice de `MQMFTCredentials.xml` por gestor de colas asociado.
4. Modalidad de compatibilidad con versiones anteriores predeterminada donde no se proporcionan detalles de credenciales para permitir la compatibilidad con releases anteriores de IBM MQo IBM WebSphere MQ

Notas:

- Los mandatos **`fteStartAgent`** y **`fteStartLogger`** no dan soporte al argumento de línea de mandatos **`-mquserid`**, ni **`-mqpassword`**, y los detalles de credenciales sólo pueden especificarse con el archivo `MQMFTCredentials.xml`.
- z/OS

  En z/OS, la contraseña debe ir en mayúsculas, aunque la contraseña del usuario tenga minúsculas. Por ejemplo, si la contraseña del usuario era "password", deberá especificarse como "PASSWORD".

Referencia relacionada

Qué mandato de MFT se conecta a qué gestor de colas
Formato del archivo de credenciales de MFT

Configuración de MQMFTCredentials.xml en Multiplatforms

Si Managed File Transfer (MFT) está configurado con la seguridad habilitada, la autenticación de conexión requiere que todos los mandatos de MFT que se conectan con un gestor de colas proporcionen credenciales de ID de usuario y contraseña. De forma similar, los registradores de MFT pueden ser necesarios para especificar un ID de usuario y una contraseña al conectarse a una base de datos. Esta información de credenciales se puede almacenar en el archivo de credenciales MFT .

Acerca de esta tarea

Los elementos del archivo `MQMFTCredentials.xml` deben ajustarse al esquema `MQMFTCredentials.xsd`. Para obtener información sobre el formato de `MQMFTCredentials.xml`, consulte Formato de archivo de credenciales MFT.

Puede encontrar un archivo de credenciales de ejemplo en el directorio `MQ_INSTALLATION_PATH/mqft/samples/credentials` .

Puede tener un archivo de credenciales de MFT para el gestor de colas de coordinación, uno para el gestor de colas de mandatos, uno para cada agente y uno para cada registrador. De forma alternativa, puede tener un archivo que sea utilizado por todo en la topología.

La ubicación predeterminada del archivo de credenciales MFT es la siguiente:

Linux AIX **AIX and Linux**

`$HOME`

Windows **Windows**

`%USERPROFILE%` o `%HOMEDRIVE%%HOMEPATH%`

Si el archivo de credenciales se almacena en una ubicación diferente, puede utilizar las propiedades siguientes para especificar dónde deben buscarlo los mandatos:

Tabla 97. : Propiedades que definen la ubicación del archivo MQMFTCredentials.xml para varios mandatos.

Tipo de mandato	Archivo de propiedades	Nombre de propiedad
Mandato que se conecta al gestor de colas de coordinación	coordination.properties	coordinationQMgrAuthenticationCredentialsFile
Mandato que se conecta al gestor de colas de mandatos	connection.properties	connectionQMgrAuthenticationCredentialsFile
Mandato que se conecta a un proceso de agente	agent.properties	agentQMgrAuthenticationCredentialsFile
Mandato que se conecta a un proceso de registrador	logger.properties	loggerQMgrAuthenticationCredentialsFile

Tabla 98. : Propiedades que definen la ubicación del archivo MQMFTCredentials.xml para agentes y procesos de registrador.

Tipo de mandato	Archivo de propiedades	Nombre de propiedad
Agentes de MFT	agent.properties	agentQMgrAuthenticationCredentialsFile
MFT registradores	logger.properties	loggerQMgrAuthenticationCredentialsFile

Para obtener detalles sobre qué mandatos y procesos se conectan a qué gestor de colas, consulte Qué mandatos y procesos de MFT se conectan a qué gestor de colas.

En lugar de añadir propiedades en archivos de propiedades individuales, puede añadir la propiedad **commonCredentialsKeyFile** al archivo `installation.properties` común existente, para que el agente, el registrador y los mandatos puedan utilizar la misma propiedad.

Puesto que el archivo de credenciales contiene información de ID de usuario y contraseña, requiere permisos especiales para impedir el acceso no autorizado al mismo:

Linux AIX **AIX and Linux**

```
chown <agent owner userid>
chmod 600
```

Windows **Windows**

Asegúrese de que la herencia no está habilitada y, a continuación, elimine todos los ID de usuario excepto los que ejecuten el agente o registrador que utilizarán el archivo de credenciales.

Los detalles de credenciales utilizados para conectarse a un gestor de colas de coordinación de MFT , en el plug-in de IBM MQ Explorer Managed File Transfer dependen del tipo de configuración:

Global (configuración en disco local)

Una configuración global utiliza el archivo de credenciales especificado en las propiedades de coordinación y mandatos.

Local (definida dentro de IBM MQ Explorer):

Una configuración local utiliza las propiedades de los detalles de conexión del gestor de colas asociado en IBM MQ Explorer.

Tareas relacionadas

“Habilitación de la autenticación de conexión para MFT” en la página 593

La autenticación de conexión del plugin de IBM MQ Explorer MFT que se conecta con un gestor de colas de coordinación o un gestor de colas de mandatos, y la autenticación de conexión para un agente de

Managed File Transfer que se conecta con un gestor de colas de coordinación o un gestor de colas de mandatos se puede ejecutar en modalidad de compatibilidad o en modalidad de autenticación MQCSP.

Crear una estructura de transferencia de archivos de IBM MQ

Referencia relacionada

Formato del archivo de credenciales de MFT

Cifrado de credenciales almacenadas en MFT

fteObfuscate: cifrar datos confidenciales

z/OS 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.xmlfile:

```
<?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

Table 99. : Properties that define the location of the MQMFTCredentials.xml file for various commands. (continued)		
Type of command	**Property file**	**Property name**
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

"Configuración de MQMFTCredentials.xml en Multiplatforms" on page 589
Si Managed File Transfer (MFT) está configurado con la seguridad habilitada, la autenticación de conexión requiere que todos los mandatos de MFT que se conectan con un gestor de colas proporcionen credenciales de ID de usuario y contraseña. De forma similar, los registradores de MFT pueden ser necesarios para especificar un ID de usuario y una contraseña al conectarse a una base de datos. Esta información de credenciales se puede almacenar en el archivo de credenciales MFT .

Habilitación de la autenticación de conexión para MFT

La autenticación de conexión del plugin de IBM MQ Explorer MFT que se conecta con un gestor de colas de coordinación o un gestor de colas de mandatos, y la autenticación de conexión para un agente de Managed File Transfer que se conecta con un gestor de colas de coordinación o un gestor de colas de mandatos se puede ejecutar en modalidad de compatibilidad o en modalidad de autenticación MQCSP.

Acerca de esta tarea

La modalidad de autenticación MQCSP es el valor predeterminado.

Para la autenticación de conexión para el plug-in de IBM MQ Explorer Managed File Transfer o para los agentes de Managed File Transfer que se conectan a un gestor de colas utilizando el transporte CLIENT, las contraseñas de más de 12 caracteres sólo están soportadas para la modalidad de autenticación MQCSP. Si especifica una contraseña de más de 12 caracteres de longitud cuando se autoriza el uso de la modalidad de compatibilidad, se produce un error y el agente no se autentica con el gestor de colas. Consulte el mensaje BFGAG0187E en la sección Mensajes de diagnóstico: BFGAG0001 - BFGAG9999.

Procedimiento

- Para seleccionar la modalidad de autenticación de conexión para un gestor de colas de coordinación o un gestor de colas de mandatos en IBM MQ Explorer, complete los pasos siguientes:
  a) Seleccione el gestor de colas al que desea conectarse.
  b) Pulse con el botón derecho del ratón y seleccione **Detalles de conexión -> Propiedades** en el menú emergente.
  c) Pulse la pestaña **ID de usuario**.
  d) Asegúrese de que el recuadro de selección para la modalidad de autenticación de conexión que desea utilizar esté seleccionado:
     – De forma predeterminada, el recuadro de selección **Modalidad de compatibilidad de identificación de usuario** no está seleccionado. Esto significa que si se selecciona el recuadro de selección **Habilitar identificación de usuario**, IBM MQ Explorer utilizará la autenticación MQCSP al conectarse al gestor de colas. Si IBM MQ Explorer debe conectarse al gestor de colas utilizando la modalidad de compatibilidad en lugar de la autenticación MQCSP, asegúrese de que estén seleccionados los recuadros de selección **Habilitar identificación de usuario** y **Modalidad de compatibilidad de identificación de usuario**.
- Para habilitar o inhabilitar la modalidad de autenticación MQCSP para un agente de Managed File Transfer utilizando el archivo `MQMFTCredentials.xml`, añada el parámetro **`useMQCSPAuthentication`** al archivo `MQMFTCredentials.xml` para el usuario pertinente.

  El parámetro **`useMQCSPAuthentication`** tiene los valores siguientes:

  true
  La modalidad de autenticación MQCSP se utiliza para autenticar el usuario con el gestor de colas.

  `true` es el valor predeterminado. Si no se especifica el parámetro **`useMQCSPAuthentication`**, se establece de forma predeterminada en `true` y la modalidad de autenticación MQCSP se utiliza para autenticar el usuario con el gestor de colas.

  falso
  La modalidad de compatibilidad se utiliza para autenticar el usuario con el gestor de colas.

En el ejemplo siguiente se muestra cómo establecer el parámetro **useMQCSPAuthentication** en el archivo `MQMFTCredentials.xml`:

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

Conceptos relacionados

"Protección por contraseña MQCSP" en la página 32
Las credenciales de autenticación que se especifican en la estructura MQCSP se pueden proteger utilizando la característica de protección de contraseña MQCSP de IBM MQ o se pueden cifrar utilizando el cifrado TLS.

Referencia relacionada

"Autenticación de conexión de MFT y IBM MQ" en la página 588
La autenticación de conexión permite que un gestor de colas se configure para autenticar aplicaciones utilizando un ID de usuario y una contraseña proporcionados. Si el gestor de colas asociado tiene la seguridad habilitada y requiere detalles de credenciales (ID de usuario y contraseña), la característica de autenticación de conexión debe estar habilitada antes de que se pueda realizar una conexión correcta a un gestor de colas. La autenticación de conexión se puede ejecutar en modalidad de compatibilidad o en modalidad de autenticación MQCSP.

Formato del archivo de credenciales de MFT

Recintos de seguridad de MFT

Puede restringir el área del sistema de archivos a la que puede acceder el agente como parte de una transferencia. El área a la que el agente está restringida se denomina el recinto de seguridad. Puede aplicar restricciones al agente o al usuario que solicita una transferencia.

Los recintos de seguridad no están soportados cuando el agente es un agente de puente de protocolo o un agente de puente Connect:Direct. No puede utilizar recintos de seguridad de agente para agentes que tienen que transferir a o desde colas de IBM MQ.

Referencia relacionada

"Trabajo con recintos de seguridad de agente MFT" en la página 594
Para añadir un nivel adicional de seguridad a Managed File Transfer, puede restringir el área de un sistema de archivos a la que un agente puede acceder.

"Trabajo con recintos de seguridad de usuario de MFT" en la página 596
Puede restringir el área del sistema de archivos de y a la que transferir los archivos dependiendo del nombre de usuario de MQMD que solicita la transferencia.

Trabajo con recintos de seguridad de agente MFT

Para añadir un nivel adicional de seguridad a Managed File Transfer, puede restringir el área de un sistema de archivos a la que un agente puede acceder.

No puede utilizar recintos de seguridad de agente para agentes que transfieren a o desde colas de IBM MQ. En su lugar, se puede implementar la restricción de acceso a las colas de IBM MQ con recintos de seguridad utilizando el recinto de seguridad de usuario que es la solución recomendada para los requisitos de recinto de seguridad. Para obtener más información sobre el recinto de seguridad de usuario, consulte "Trabajo con recintos de seguridad de usuario de MFT" en la página 596

Para habilitar el recinto de seguridad de agente, añada la siguiente propiedad al archivo `agent.properties` del agente que desea restringir:

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

donde:

- *restricted_directory_name* es una vía de acceso de directorio que se debe permitir o denegar.

- ! es opcional y especifica que se deniega (excluye) el siguiente valor para *restricted_directory_name* . Si no se especifica ! , *restricted_directory_name* es una vía de acceso permitida (incluida).
- *separator* es el separador específico de la plataforma.

Por ejemplo, si sólo desea restringir el acceso que AGENT1 tiene al directorio `/tmp`, pero no desea permitir el acceso al subdirectorio `private`, establezca la propiedad de la siguiente manera en el archivo `agent.properties` perteneciente a AGENT1: `sandboxRoot=/tmp:!/tmp/private`.

La propiedad sandboxRoot se describe en Propiedades avanzadas del agente.

No se soporta la creación de recintos de seguridad de usuario y de agente en los agentes de puente de protocolo o en agentes de puente de Connect:Direct.

Cómo trabajar en un recinto de seguridad en plataformas AIX, Linux, and Windows

ALW En plataformas AIX, Linux, and Windows , el recinto de seguridad restringe qué directorios puede leer y escribir un Managed File Transfer Agent . Cuando se activa el recinto de seguridad, el Managed File Transfer Agent puede leer y escribir en los directorios en los que se permite y en sus subdirectorios a menos que estos estén especificados como denegados en sandboxRoot. El recinto de seguridad de Managed File Transfer no tiene prioridad sobre la seguridad del sistema operativo. El usuario que ha iniciado el Managed File Transfer Agent debe tener el acceso de nivel de sistema operativo adecuado a cualquier directorio para poder leer o escribir en dicho directorio. No se sigue un enlace simbólico a un directorio si el directorio enlazado está fuera de los directorios (y subdirectorios) sandboxRoot.

Trabajar en un recinto de seguridad en z/OS

z/OS En z/OS, el recinto de seguridad restringe los cualificadores de nombre de conjunto de datos en los que el Managed File Transfer Agent puede leer y escribir. El usuario que ha iniciado el Managed File Transfer Agent debe tener las correspondientes autorizaciones del sistema operativo para los conjuntos de datos implicados. Si escribe un valor de calificador de nombre de conjunto de datos sandboxRoot entre comillas dobles, el valor sigue el convenio normal de z/OS y se trata como un valor totalmente calificado. Si omite las comillas dobles, al directorio sandboxRoot se le antepone como prefijo el ID de usuario actual. Por ejemplo, si establece la propiedad sandboxRoot en lo siguiente: `sandboxRoot=//test`, el agente puede acceder a los siguientes conjuntos de datos (en notación z/OS estándar) `//`*`username`*`.test.**` En tiempo de ejecución, si los niveles iniciales del nombre de conjunto de datos totalmente resuelto no coinciden con el sandboxRoot, se rechaza la solicitud de transferencia.

Trabajar en un recinto de seguridad en sistemas IBM i

IBM i En los archivos del sistema de archivos integrado en sistemas IBM i, el recinto de seguridad restringe los directorios en los que un Managed File Transfer Agent puede leer y escribir. Cuando se activa el recinto de seguridad, el Managed File Transfer Agent puede leer y escribir en los directorios en los que se permite y en sus subdirectorios a menos que estos estén especificados como denegados en sandboxRoot. El recinto de seguridad de Managed File Transfer no tiene prioridad sobre la seguridad del sistema operativo. El usuario que ha iniciado el Managed File Transfer Agent debe tener el acceso de nivel de sistema operativo adecuado a cualquier directorio para poder leer o escribir en dicho directorio. No se sigue un enlace simbólico a un directorio si el directorio enlazado está fuera de los directorios (y subdirectorios) sandboxRoot.

Referencia relacionada

“Comprobaciones adicionales de transferencias de comodín” en la página 599
Si se ha configurado un agente con un recinto de pruebas de usuario o agente para poder restringir las ubicaciones en las que el agente puede transferir archivos, y puede especificar que comprobaciones adicionales se van a realizar en transferencias de comodín para dicho agente.

“Trabajo con recintos de seguridad de agente MFT” en la página 594
Para añadir un nivel adicional de seguridad a Managed File Transfer, puede restringir el área de un sistema de archivos a la que un agente puede acceder.

Trabajo con recintos de seguridad de usuario de MFT

Puede restringir el área del sistema de archivos de y a la que transferir los archivos dependiendo del nombre de usuario de MQMD que solicita la transferencia.

Los recintos de seguridad de usuario no están soportados cuando el agente es un agente de puente de protocolo o un agente de puente Connect:Direct.

Para habilitar los recintos de seguridad, añada la siguiente propiedad al archivo agent.properties para el agente que desea restringir:

```
userSandboxes=true
```

Cuando esta propiedad está presente y se establece en true, el agente utiliza la información del archivo *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/agents/*agent_name*/UserSandboxes.xml para determinar a qué partes del sistema de archivos puede acceder el usuario que solicita la transferencia.

El XML UserSandboxes.xml se compone de un elemento <agent> que contiene cero o más elementos <sandbox> . Estos elementos describen qué reglas se aplican a qué usuarios. El atributo user del elemento <sandbox> es un patrón que se utiliza para buscar coincidencias con el usuario MQMD de la solicitud.

El agente vuelve a cargar periódicamente el archivo UserSandboxes.xml y cualquier cambio válido en el archivo afectará al comportamiento del agente. El intervalo de recarga predeterminado es de 30 segundos. Este intervalo se puede cambiar especificando la propiedad de agente xmlConfigReloadInterval en el archivo agent.properties.

Si se especifica el atributo o valor userPattern="regex", el atributo user se interpreta como una expresión regular Java. Si desea más información, consulte Expresiones regulares utilizadas por MFT.

Si no especifica el atributo o valor userPattern="regex", el atributo user se interpreta como un patrón con los siguientes caracteres comodín:

- asterisco (*), que representa cero o más caracteres
- signo de interrogación (?), que representa exactamente un carácter

Las coincidencias se realizan en el orden en el que los elementos <sandbox> se listan en el archivo. Sólo se utiliza la primera coincidencia, todas las siguientes coincidencias potenciales en el archivo se ignoran. Si ninguno de los elementos <sandbox> especificados en el archivo coincide con el usuario MQMD asociado con el mensaje de solicitud de transferencia, la transferencia no puede acceder al sistema de archivos. Cuando se encuentra una coincidencia entre el nombre de usuario MQMD y un atributo user, la coincidencia identifica un conjunto de reglas dentro de un elemento <sandbox> que se aplican a la transferencia. Este conjunto de reglas se utiliza para determinar qué archivos, o conjuntos de datos, pueden leerse o escribirse como parte de la transferencia.

Cada conjunto de reglas puede especificar un elemento <read> , que identifica qué archivos se pueden leer, y un elemento <write> que identifica qué archivos se pueden escribir. Si omite los elementos <read> o <write> de un conjunto de reglas, se supone que el usuario asociado con dicho conjunto de reglas no tiene permiso para realizar ninguna lectura ni escritura, según corresponda.

Nota: El elemento <read> debe estar antes del elemento <write>, y el elemento <include> debe ser anterior al elemento <exclude>, en el archivo UserSandboxes.xml.

Cada elemento <read> o <write> contiene uno o más patrones que se utilizan para determinar si un archivo está en el recinto de seguridad y se puede transferir. Especifique estos patrones utilizando los elementos <include> y <exclude>. El atributo name del elemento <include> o <exclude> especifica el patrón que debe coincidir. Un atributo type opcional especifica si el valor de nombre es un

patrón de cola o un archivo. Si no se especifica el atributo `type`, el agente trata el patrón como un patrón de vía de acceso de archivo o de directorio. Por ejemplo:

```
<tns:read>
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

El agente utiliza los patrones `<include>` y `<exclude>` `name` para determinar si los archivos, conjuntos de datos o colas se pueden leer o grabar en ellos. Se permite una operación si el nombre canónico de la vía de acceso de archivo, del conjunto de datos o de la cola coincide con al menos uno de los patrones incluidos y exactamente cero de los patrones excluidos. Los patrones especificados utilizando el atributo `name` de los elementos `<include>` y `<exclude>` utilizan los convenios y separadores de vía de acceso correspondientes a la plataforma en que se está ejecutando el agente. Si especifica vías de acceso relativas, las vías de acceso serán resueltas en la propiedad `transferRoot` del agente.

Cuando se especifica una restricción de cola, una sintaxis de QUEUE@QUEUEMANAGER está soportada, con las reglas siguientes:

- Si falta el carácter de arroba (@) en la entrada, el patrón se trata como un nombre de cola al que se puede acceder a cualquier gestor de colas. Por ejemplo, si el patrón es `name`, se trata de la misma manera que `name@**`.
- Si el carácter de arroba (@) es el primer carácter de la entrada, el patrón se trata como un nombre de gestor de colas y se puede acceder a todas las colas del gestor de colas. Por ejemplo, si el patrón es `@name`, se trata de la misma manera que `**@name`..

Los siguientes caracteres comodín tienen un significado especial cuando se especifican como parte del atributo `name` de los elementos `<include>` y `<exclude>`:

*****
: Un único asterisco coincide con cero o más caracteres en un nombre de directorio, o en un calificador de un nombre de conjunto de datos o nombre de cola .

?
: Un signo de interrogación coincide exactamente con un carácter en un nombre de directorio, o en un calificador de un nombre de conjunto de datos o nombre de cola.

******
: Dos caracteres de asterisco coinciden con cero o más nombres de directorio, o cero o más calificadores en un nombre de conjunto de datos de o nombre de cola de . Además las vías de acceso que finalizan con un separador de vía de acceso tienen dos asteriscos "**" implícitos añadidos al final de la vía de acceso. Por lo tanto, `/home/user/` es el mismo que `/home/user/**`.

Por ejemplo:

- `/**/test/**` coincide con cualquier archivo con un directorio `test` en su vía de acceso
- `/test/file?` coincide con cualquier archivo del directorio `/test` que empiece por la serie `file` seguido de cualquier carácter único
- `c:\test\*.txt` coincide con cualquier archivo dentro del directorio `c:\test` con una extensión `.txt`
- `c:\test\**\*.txt` coincide con cualquier archivo dentro del directorio `'c:\test` o uno de sus subdirectorios con una extensión `.txt`
- z/OS `//'TEST.*.DATA'` coincide con cualquier conjunto de datos que tenga el primer calificador TEST, tiene cualquier segundo calificador y un tercero de DATA.
- *@QM1 coincide con cualquier cola del gestor de colas QM1 que tenga un único calificador.
- TEST.*.QUEUE@QM1 coincide con cualquier cola del gestor de colas QM1 que tiene el primer cualificador de TEST, tiene cualquier segundo cualificador y un tercero de QUEUE.
- **@QM1 coincide con cualquier cola del gestor de colas QM1.

Enlaces simbólicos

Debe resolver por completo los enlaces simbólicos que se utilizan en las vías de acceso de archivo en el archivo `UserSandboxes.xml` especificando enlaces fijos en los elementos `<include>` y `<exclude>`. Por ejemplo, si tiene un enlace simbólico donde `/var` se correlaciona con `/SYSTEM/var`, debe especificar esta vía de acceso como `<tns:include name="/SYSTEM/var"/>`, de lo contrario la transferencia prevista fallará con un error de seguridad de recinto de seguridad de usuario.

Ejemplo

Este ejemplo muestra cómo permitir que el usuario con el nombre de usuario MQMD `guest` transfiera cualquier archivo desde el directorio `/home/user/public` o cualquiera de sus subdirectorios en el sistema donde se ejecuta el agente AGENT_JUPITER, añadiendo el siguiente elemento `<sandbox>` al archivo `UserSandboxes.xml` en el directorio de configuración de 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>
```

Ejemplo

En este ejemplo se muestra cómo permitir que cualquier usuario con el nombre de usuario MQMD `account` seguido de un único dígito, por ejemplo, `account4`, complete las siguientes acciones:

- Transfiera cualquier archivo desde el directorio `/home/account` o cualquiera de sus subdirectorios, excluyendo el directorio `/home/account/private` en el sistema donde se está ejecutando el agente AGENT_SATURN
- Transfiera cualquier archivo al directorio `/home/account/output` o a cualquiera de sus subdirectorios en el sistema donde se está ejecutando AGENT_SATURN
- Leer mensajes de las colas del gestor de colas local que empiezan por el prefijo `ACCOUNT.` a menos que empiece por `ACCOUNT.PRIVATE.` (es decir, que tenga `PRIVATE` en el segundo nivel).
- Transfiera datos a las colas que empiezan con el prefijo `ACCOUNT.OUTPUT.` en cualquier gestor de colas.

Para permitir que un usuario con el nombre de usuario MQMD `account` complete estas acciones, añada el siguiente elemento `<sandbox>` al archivo `UserSandboxes.xml`, en el directorio de configuración de 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>
```

Referencia relacionada

“Comprobaciones adicionales de transferencias de comodín” en la página 599
Si se ha configurado un agente con un recinto de pruebas de usuario o agente para poder restringir las ubicaciones en las que el agente puede transferir archivos, y puede especificar que comprobaciones adicionales se van a realizar en transferencias de comodín para dicho agente.

El archivo MFT `agent.properties`

Comprobaciones adicionales de transferencias de comodín

Si se ha configurado un agente con un recinto de pruebas de usuario o agente para poder restringir las ubicaciones en las que el agente puede transferir archivos, y puede especificar que comprobaciones adicionales se van a realizar en transferencias de comodín para dicho agente.

Propiedad additionalWildcardSandboxChecking

Para habilitar una comprobación adicional de transferencias de comodín, añada la siguiente propiedad al archivo `agent.properties` para el agente que desea comprobar.

```
additionalWildcardSandboxChecking=true
```

Si esta propiedad está establecida en true y el agente realiza una solicitud de transferencia que intenta leer una ubicación que está fuera del recinto de seguridad definido para la coincidencia de archivos del comodín, la transferencia falla. Si hay varias transferencias dentro de una solicitud de transferencia, y una de estas solicitudes falla debido a que intenta leer una ubicación fuera del recinto de seguridad, toda la transferencia falla. Si la comprobación falla, la razón del fallo se proporciona en un mensaje de error.

Si se omite la propiedad additionalWildcardSandboxChecking property del archivo `agent.properties` de un agente o se establece en false, no se realiza ninguna comprobación adicional en las transferencias de comodín para ese agente.

Mensajes de error para la comprobación de comodín

Los mensajes que se notifican cuando se realiza una solicitud de transferencia de comodín a una ubicación fuera de una ubicación de recinto de seguridad configurada son los siguientes.

Aparece el mensaje siguiente cuando una vía de acceso de archivo de comodín en una solicitud de transferencia se encuentra fuera del recinto de seguridad restringido.

```
BFGSS0077E: Se ha rechazado el intento de leer la vía de acceso de archivo vía de acceso.
La vía de acceso del archivo está situada fuera del recinto de seguridad de transferencias
restringido.
```

Aparece el mensaje siguiente cuando una transferencia dentro de una solicitud de transferencia múltiple contiene una solicitud de transferencia de comodín donde la vía de acceso se encuentra fuera del recinto de seguridad restringido.

```
BFGSS0078E: Se ha hecho caso omiso del intento de leer la vía de acceso de archivo: vía de acceso
como otra transferencia.
El elemento de la transferencia gestionada ha intentado leer fuera del recinto de seguridad de
transferencias restringido.
```

Aparece el mensaje siguiente cuando un archivo se encuentra fuera del recinto de seguridad restringido:

```
BFGSS0079E: Se ha denegado el intento de leer el archivo vía de acceso de archivo.
El archivo está situado fuera del recinto de seguridad de transferencias restringido.
```

Aparece el mensaje siguiente en una solicitud de transferencia múltiple donde otra solicitud de transferencia de comodín ha causado que se hiciera caso omiso de esta:

```
BFGSS0079E: Se ha hecho caso omiso del intento de leer el archivo: vía de acceso de archivo como
otra transferencia.
El elemento de la transferencia gestionada ha intentado leer fuera del recinto de seguridad de
transferencias restringido.
```

En el caso de transferencias de archivos únicas que no incluyen caracteres comodín, el mensaje notificado cuando la transferencia implica que un archivo que se encuentra fuera del recinto de seguridad no se ha modificado en releases anteriores:

```
Errores con BFGIO0056E: se ha denegado el intento de leer el archivo "ARCHIVO".
El archivo está situado fuera del recinto de seguridad de transferencias restringido.
```

Referencia relacionada

"Trabajo con recintos de seguridad de usuario de MFT" en la página 596
Puede restringir el área del sistema de archivos de y a la que transferir los archivos dependiendo del nombre de usuario de MQMD que solicita la transferencia.

"Trabajo con recintos de seguridad de agente MFT" en la página 594
Para añadir un nivel adicional de seguridad a Managed File Transfer, puede restringir el área de un sistema de archivos a la que un agente puede acceder.

El archivo MFT `agent.properties`

Configurar el cifrado SSL o TLS para MFT

Puede utilizar SSL o TLS con IBM MQ Managed File Transfer para proteger la comunicación entre los agentes y sus gestores de colas de agente, los mandatos y los gestores de colas a los que se están conectando, y las diversas conexiones de gestor de colas con el gestor de colas dentro de la topología.

Antes de empezar

Puede utilizar el cifrado SSL o TLS para cifrar los mensajes que fluyen a través de una topología de IBM MQ Managed File Transfer . Incluyen los siguientes:

- Mensajes que pasan entre un agente y su gestor de colas de agente.
- Mensajes para los mandatos y los gestores de colas a los que se están conectando.
- Mensajes internos que fluyen entre los gestores de colas de agente, los gestores de colas de mandatos y el gestor de colas de coordinación dentro de la topología.

Acerca de esta tarea

Si desea información general sobre cómo utilizar SSL con IBM MQ, consulte "Trabajar con SSL/TLS" en la página 281. Desde el punto de vista de IBM MQ, Managed File Transfer es una aplicación de cliente Java estándar.

Siga estos pasos para utilizar SSL con Managed File Transfer:

Procedimiento

1. Cree un archivo de almacén de confianza y, opcionalmente, un archivo de almacén de claves (estos archivos pueden ser el mismo archivo). Si no necesita autenticación de cliente (es decir, SSLCAUTH=OPTIONAL en canales) no necesita proporcionar un almacén de claves. Sólo necesita un almacén de confianza para autenticar el certificado del gestor de colas.

   El algoritmo de clave utilizado para crear certificados para el almacén de confianza y los almacenes de claves debe ser RSA para trabajar con IBM MQ.
2. Configure el gestor de colas de IBM MQ para que utilice SSL.

   Para obtener información sobre cómo configurar un gestor de colas para que utilice SSL mediante IBM MQ Explorer, por ejemplo, consulte Configurar SSL en los gestores de colas.
3. Guarde el archivo de almacén de confianza y el archivo de almacén de claves (si dispone de uno) en una ubicación adecuada. Una ubicación sugerida es el directorio *config_directory*/*coordination_qmgr*/agents/*agent_name* .
4. Establezca las propiedades SSL según sea necesario para cada gestor de colas habilitado para SSL en el archivo de propiedades de Managed File Transfer adecuado. Cada conjunto de propiedades se refiere a un gestor de colas separado (agente, coordinación y mandato) aunque un gestor de colas puede ejecutar dos o más de estos roles.

Se precisa una de las propiedades **CipherSpec** o **CipherSuite**, de lo contrario, el cliente intente conectarse sin SSL. Se suministran las propiedades **CipherSpec** y **CipherSuite** debido a las diferencias terminológicas entre IBM MQ y Java. Managed File Transfer acepta la propiedad y no efectúa la conversión necesaria, para que no necesite establecer ambas propiedades. Si especifica las propiedades **CipherSpec** o **CipherSuite**, **CipherSpec** tiene prioridad.

El parámetro **PeerName** es opcional. Puede establecer la propiedad en el nombre distinguido del gestor de colas al que se desea conectar. Managed File Transfer rechaza las conexiones a un servidor SSL incorrecto con un nombre distinguido que no coincida.

Establezca las propiedades **SslTrustStore** y **SslKeyStore** en nombres de archivo que apunten a los archivos de almacén de confianza y de almacén de claves. Si está estableciendo estas propiedades para un agente que ya está en ejecución, detenga y reinicie el agente para reconectarse en modalidad SSL.

Los archivos de propiedades contienen contraseñas de texto sin formato; por consiguiente, contemple a posibilidad de otorgar los permisos correspondientes a los sistemas de archivos.

Para obtener más información sobre las propiedades de SSL, consulte "Propiedades SSL/TLS para MFT" en la página 601.

5. Si un gestor de colas de agente utiliza SSL, no puede proporcionar los detalles necesarios cuando crea el agente. Para crear el agente, efectúe los pasos siguientes:
   a) Cree el agente utilizando el mandato **fteCreateAgent**. Recibirá un aviso sobre la imposibilidad de publicar la existencia del agente en el gestor de colas de coordinación.
   b) Edite el archivo `agent.properties` que se creó mediante el paso anterior para añadir la información SSL. Cuando el agente se ha iniciado correctamente, se vuelve a intentar la publicación.
6. Si hay agentes o instancias de IBM MQ Explorer en ejecución mientras se modifican las propiedades SSL del archivo `agent.properties` o el archivo `coordination.properties`, debe reiniciar el agente o IBM MQ Explorer.

Referencia relacionada

El archivo MFT `agent.properties`

Propiedades SSL/TLS para MFT

Algunos archivos de propiedades MFT incluyen propiedades SSL y TLS. Puede utilizar SSL o TLS con IBM MQ y Managed File Transfer para evitar conexiones no autorizadas entre agentes y gestores de colas, y para cifrar el tráfico de mensajes entre agentes y gestores de colas.

Los siguientes archivos de propiedades MFT incluyen propiedades SSL:

- Propiedades SSL/TLS para el archivo MFT `agent.properties`
- Propiedades SSL/TLS para el archivo MFT `coordination.properties`
- Propiedades SSL/TLS para el archivo MFT `command.properties`
- Propiedades SSL/TLS para el archivo MFT `logger.properties`

Para obtener información sobre cómo utilizar SSL o TLS con Managed File Transfer, consulte "Configurar el cifrado SSL o TLS para MFT" en la página 600.

A partir de IBM WebSphere MQ 7.5, puede utilizar variables de entorno en algunas propiedades de Managed File Transfer que representan ubicaciones de archivo o directorio. Esto permite que las ubicaciones de archivos o directorios que se utilizan al ejecutar componentes del producto varíen en función de los cambios de entorno, por ejemplo en función del usuario que esté ejecutando el proceso. Para obtener más información, consulte El uso de variables de entorno en las propiedades de MFT.

Conceptos relacionados

Opciones de configuración de MFT en Multiplatforms

Referencia relacionada

El uso de variables de entorno en las propiedades de MFT

Conexión a un gestor de colas en modalidad de cliente con autenticación de canal

IBM MQ utiliza registros de autenticación de canal para controlar de forma más precisa el acceso a nivel de canal. Esto significa que, de forma predeterminada, los gestores de colas recién creados rechazan las conexiones de cliente del componente Managed File Transfer .

Si desea más información sobre la autenticación de canal, consulte "Registros de autenticación de canal" en la página 53.

Si la configuración de autenticación de canal del SVRCONN utilizado por Managed File Transfer especifica un ID MCAUSER sin privilegios, hay que otorgar registros de autorización específicos para el gestor de colas, las colas y los temas para que los mandatos y el Managed File Transfer Agent funcionen correctamente. Utilice el mandato de MQSC SET CHLAUTH o el mandato de PCF Establecer registro de autenticación de canal para crear, modificar o eliminar registros de autenticación de canal. Para todos los agentes de Managed File Transfer que desee conectar al gestor de colas de IBM MQ , puede configurar un ID MCAUSER para utilizarlo para todos los agentes, o configurar un ID MCAUSER independiente para cada agente.

Otorgue a cada ID MCAUSER los permisos siguientes:

- Registros de autorización necesarios para el gestor de colas:
  - conectar
  - setid
  - inq
- Registros de autorización necesarios para colas.

  Para todas las colas específicas del agente, es decir, los nombres de cola que finalizan en *nombre_agente* en la lista siguiente, debe crear estos registros de autorización de cola para cada agente que desee conectar con el gestor de colas IBM MQ utilizando una conexión de cliente.
  - put, get, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)
  - put, get, setid, browse (SYSTEM.FTE.COMMAND.*nombre_agente*)
  - put, get (SYSTEM.FTE.DATA.*nombre_agente*)
  - put, get (SYSTEM.FTE.REPLY.*nombre_agente*)
  - put, get, inq, browse (SYSTEM.FTE.STATE.*nombre_agente*)
  - put, get, browse (SYSTEM.FTE.EVENT.*nombre_agente*)
  - put, get (SYSTEM.FTE)
- Registros de autorización necesarios para temas:
  - sub, pub (SYSTEM.FTE)
- Se requieren registros de autorización para las transferencias de archivos.

  Si tiene ID de MCAUSER separados para el agente de origen y de destino, cree los registros de autorización en las colas de los agentes de origen y de destino.

  Por ejemplo, si el ID de MCAUSER del agente de origen es **user1** y el ID de MCAUSER del agente de destino es **user2**, establezca las siguientes autorizaciones para los usuarios del agente:

Usuario AGENT	Cola	Autorización necesaria
user1	SYSTEM.FTE.DATA.*nombre_agente_destino*	put
user1	SYSTEM.FTE.COMMAND.*nombre_agente_destino*	put
user2	SYSTEM.FTE.REPLY.*nombre_agente_origen*	put
user2	SYSTEM.FTE.COMMAND.*nombre_agente_origen*	put

Configuración de SSL o TLS entre el agente de puente Connect:Direct y el nodo Connect:Direct

Configure el agente de puente Connect:Direct y el nodo Connect:Direct para conectarse entre sí a través del protocolo SSL, creando un almacén de claves y un almacén de confianza y estableciendo propiedades en el archivo de propiedades del agente de puente Connect:Direct.

Acerca de esta tarea

Estos pasos incluyen instrucciones para recibir las claves firmadas por una entidad emisora de certificados. Si no utiliza una entidad emisora de certificados, puede generar un certificado autofirmado. Si desea más información sobre cómo generar un certificado firmado automáticamente, consulte "Trabajar con SSL/TLS en AIX, Linux, and Windows" en la página 299.

Estos pasos incluyen instrucciones para crear un nuevo almacén de claves y un nuevo almacén de confianza para el agente de puente Connect:Direct. Si el agente de puente Connect:Direct ya tiene un almacén de claves y un almacén de confianza que utiliza para conectarse de forma segura a gestores de colas de IBM MQ, puede utilizar el almacén de claves y el almacén de confianza existentes al conectarse de forma segura al nodo Connect:Direct. Para obtener más información, consulte "Configurar el cifrado SSL o TLS para MFT" en la página 600.

Procedimiento

Para el nodo Connect:Direct, realice los pasos siguientes:

1. Genere una clave y un certificado firmado para el nodo Connect:Direct.

   Puede hacer esto con la herramienta IBM Key Management que se proporciona con IBM MQ. Para obtener más información, consulte "Trabajar con SSL/TLS" en la página 281.
2. Envíe una solicitud a una entidad emisora de certificados para que le firme la clave. Recibirá a cambio un certificado.
3. Cree un archivo de texto, por ejemplo `/test/ssl/certs/CAcert`, que contenga la clave pública de la entidad emisora de certificados.
4. Instale la Opción Secure+ en el nodo Connect:Direct.

   Si el nodo ya existe, puede instalar la Opción Secure+ ejecutando de nuevo el instalador, especificando la ubicación de la instalación existente y eligiendo instalar sólo la Opción Secure+.
5. Cree un archivo de texto nuevo; por ejemplo, `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`.
6. Copie el certificado que ha recibido de la autoridad de certificación y la clave privada, que se encuentra en `/test/ssl/cd/privateKeys/`*`node_name`*`.key`, en el archivo de texto.

   El contenido de `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt` debe estar en el formato siguiente:

```
-----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 la Herramienta de administración Secure+.
   - En sistemas AIX and Linux , ejecute el mandato **spadmin.sh**.
   - En sistemas Windows, pulse **Inicio** > **Programas** > **Sterling Commerce Connect: Direct** > **Herramienta de administración CD Secure+**

   Se inicia la Herramienta de administración CD Secure+.
8. En la Herramienta de administración CD Secure+, efectúe una doble pulsación en la línea **.Local** para editar la configuración SSL o TLS principal.
   a) Seleccione **Habilitar protocolo SSL** o **Habilitar protocolo TLS**, dependiendo del protocolo que esté utilizando.
   b) Seleccione **Inhabilitar alteración temporal**.
   c) Seleccione al menos una suite de cifrado.
   d) Si desea autenticación bidireccional, cambie el valor de **Habilitar autenticación de cliente** a `Yes`.
   e) En el campo **Certificado raíz de confianza**, especifique la vía de acceso al archivo de certificado público de la entidad emisora de certificados, `/test/ssl/certs/CAcert`.
   f) En el campo **Archivo de certificado de clave** , especifique la vía de acceso al archivo que ha creado, `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`.
9. Efectúe una doble pulsación en la línea **.Client** para editar la configuración SSL o TLS principal.
   a) Seleccione **Habilitar protocolo SSL** o **Habilitar protocolo TLS**, dependiendo del protocolo que esté utilizando.
   b) Seleccione **Inhabilitar alteración temporal**.

Para el agente de puente Connect:Direct, realice los pasos siguientes:

10. Cree un almacén de confianza. Puede hacer esto creando una clave ficticia y suprimiendo luego la clave ficticia.

    Puede utilizar los siguientes mandatos:

    ```
    keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
    ```

    ```
    keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
    ```

11. Importe el certificado público de la entidad emisora de certificados al almacén de confianza.

    Puede utilizar el siguiente mandato:

    ```
    keytool -import -trustcacerts -alias myCA
            -file /test/ssl/certs/CAcert
            -keystore /test/ssl/fte/stores/truststore.jks
    ```

12. Edite el archivo de propiedades del agente de puente Connect:Direct.

    Incluya las siguientes líneas en cualquier parte del archivo:

    ```
    cdNodeProtocol=protocol
    ```

```
cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
cdNodeTruststorePassword=password
```

En el ejemplo de este paso, *protocolo* es el protocolo que está utilizando, ya sea SSL o TLS, y *contraseña* es la contraseña que especificó cuando creó el almacén de confianza.

13. Si desea autenticación bidireccional, cree una clave y un certificado para el agente de puente Connect:Direct.
    a) Cree un almacén de claves y una clave.

    Puede utilizar el siguiente mandato:

    ```
    keytool -genkey -keyalg RSA -alias agent_name
            -keystore /test/ssl/fte/stores/keystore.jks
            -storepass password -validity 365
    ```

    b) Genere una solicitud de firma.

    Puede utilizar el siguiente mandato:

    ```
    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 el certificado que reciba del paso anterior al almacén de claves. El certificado debe estar en formato x.509.

    Puede utilizar el siguiente mandato:

    ```
    keytool -import -keystore /test/ssl/fte/stores/keystore.jks
            -storepass password -file certificate_file_path
    ```

    d) Edite el archivo de propiedades del agente de puente Connect:Direct.

    Incluya las siguientes líneas en cualquier parte del archivo:

    ```
    cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
    cdNodeKeystorePassword=password
    ```

    En el ejemplo de este paso, *contraseña* es la contraseña que especificó cuando creó el almacén de claves.

Tareas relacionadas

Configurar el puente Connect:Direct

ALW Protección de clientes de AMQP

Puede utilizar una serie de mecanismos de seguridad para proteger las conexiones de los clientes AMQP y asegurarse de que los datos están protegidos adecuadamente en la red. Puede crear seguridad en las aplicaciones MQ Light . También puede utilizar las funciones de seguridad existentes de IBM MQ con clientes AMQP, de la misma forma que se utilizan las características para otras aplicaciones.

Reglas de autenticación de canal (CHLAUTH)

Puede utilizar las reglas de autenticación de canal para restringir las conexiones TCP al gestor de colas. Los canales AMQP admiten el uso de reglas de autenticación de canal que puede configurar para el gestor de colas. Si las reglas de autenticación de canal se definen con un perfil que coincide con los canales AMQP en el gestor de colas, estas reglas se aplican a los canales. De forma predeterminada, la autenticación de canal está habilitada en los nuevos gestores de colas de IBM MQ por lo que debe completar al menos alguna configuración antes de poder utilizar un canal AMQP.

Para obtener más información sobre cómo configura reglas de autenticación de canal para permitir conexiones AMQP al gestor de colas, consulte Creación y utilización de canales AMQP.

Autenticación de conexión (CONNAUTH)

Puede utilizar la autenticación de conexión para autenticar las conexiones a un gestor de colas. Los canales AMQP admiten el uso de la autenticación de conexión para controlar el acceso al gestor de colas de las aplicaciones AMQP.

El protocolo AMQP utiliza la infraestructura SASL (capa de seguridad y autenticación simple) para especificar cómo se autentica una conexión. Hay varios mecanismos SASL y IBM MQ admite dos mecanismos SASL: ANONYMOUS y PLAIN.

En el caso de ANONYMOUS, no se pasa ninguna credencial del cliente al gestor de colas para la autenticación. Si el objeto IBM MQ AUTHINFO que se especifica en el atributo **CONNAUTH** del gestor de colas tiene un valor **CHCKCLNT** de REQUIRED o REQDADM (si se conecta como usuario administrativo), se rechaza la conexión. Si el valor de **CHCKCLNT** es NONE o OPTIONAL, se acepta la conexión.

En el caso de PLAIN, se pasa el nombre de usuario y la contraseña del cliente al gestor de colas para la autenticación. Si el objeto IBM MQ AUTHINFO especificado en el atributo **CONNAUTH** del gestor de colas tiene un valor **CHCKCLNT** de NONE, la conexión se rechaza. Si el valor de **CHCKCLNT** es OPTIONAL, REQUIREDo REQDADM (si se conecta como usuario administrativo), el nombre de usuario y la contraseña los comprueba el gestor de colas. El gestor de colas comprueba el sistema operativo (si el objeto AUTHINFO es de tipo IDPWOS) o un repositorio LDAP (si el objeto AUTHINFO es de tipo IDPWLDAP).

En la tabla siguiente se resume este comportamiento de autenticación:

Tabla 101. Resumen de los mecanismos SASL y la autenticación de conexión

Mecanismo SASL	**¿Se pasan las credenciales del cliente al gestor de colas?**	**Valor de CHKCLNT**
ANONYMOUS	No	REQUIRED o REQDADM: se rechaza la conexión NONE o OPTIONAL: se acepta la conexión
PLAIN	Sí, nombre de usuario y la contraseña	REQUIRED REQDADM u OPTIONAL: el gestor de colas comprueba el nombre de usuario y la contraseña NONE: se rechaza la conexión

Si utiliza un cliente de MQ Light, puede especificar las credenciales incluyéndolas en la dirección de AMQP a la que se conecta, por ejemplo:

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

Valor de MCAUSER en un canal

Los canales AMQP tienen un atributo MCAUSER, que puede utilizar para establecer el ID de usuario de IBM MQ bajo el que todas las conexiones a dicho canal tienen autorización. Todas las conexiones de clientes AMQP a dicho canal adoptan el ID de MCAUSER configurado. Ese ID de usuario se utiliza para la autorización de mensajería en diferentes temas.

Se recomienda utilizar la autenticación de canal (CHLAUTH) para proteger las conexiones a gestores de colas. Si está utilizando la autenticación de canal, se recomienda configurar el valor de MCAUSER para un usuario sin privilegios. Así se garantiza que si una conexión a un canal no coincide con una regla CHLAUTH, no se autoriza la conexión a realizar operaciones de mensajería en el gestor de colas.

soporte SSL/TLS

Los canales AMQP admiten el cifrado SSL/TLS utilizando claves del repositorio de claves configurado para el gestor de colas. Las opciones de configuración de canal AMQP para el cifrado SSL/TLS admiten las mismas opciones que otros tipos de canal MQ; puede indicar una especificación de cifrado y si el gestor de colas requiere certificados de las conexiones de cliente AMQP.

Mediante el uso de atributos FIPS del gestor de colas puede controlar las suites de cifrado SSL/TLS, que puede utilizar para proteger las conexiones de clientes AMQP.

Para obtener información sobre cómo configurar un repositorio de claves para el gestor de colas, consulte "Trabajar con SSL/TLS en AIX, Linux, and Windows" en la página 299.

Para obtener información sobre cómo configurar el soporte SSL/TLS para una conexión de cliente AMQP, consulte Creación y utilización de canales AMQP.

V 9.4.0 V 9.4.0 A partir de IBM MQ 9.4.0, el canal AMQP ya no da soporte a los repositorios de claves CMS en el gestor de colas. Puede utilizar el mandato **runmqakm** para convertir un repositorio de claves CMS al formato PKCS #12 , que está soportado. Por ejemplo, puede utilizar el mandato siguiente para convertir un repositorio de claves denominado `sslTest.kdb` del formato CMS al formato PKCS #12 . El nuevo repositorio de claves se denomina `sslTest.p12`y está protegido con la contraseña `passw0rd`.

```
runmqakm -keydb -convert -type cms -db sslTest.kdb -stashed -new_format pkcs12 -target
sslTest.p12 -new_pw passw0rd
```

Java Authentication and Authorization Service (JAAS)

Puede configurar opcionalmente canales AMQP con un módulo de inicio de sesión JAAS, que puede comprobar el nombre de usuario y la contraseña proporcionados por un cliente AMQP. Consulte "Configuración de JAAS para canales AMQP" en la página 608.

Tareas relacionadas

Desarrollo de aplicaciones cliente AMQP

Creación y utilización de canales AMQP

ALW Restricción de la toma de control del cliente AMQP

Cuando se realiza una conexión de cliente AMQP que tiene el mismo identificador de cliente que una conexión de cliente AMQP existente, la conexión de cliente existente se desconecta de forma predeterminada. Sin embargo, puede configurar el gestor de colas para restringir el comportamiento de toma de control del cliente para que la toma de control sea posible solo cuando se cumplan determinados criterios.

Por ejemplo, es posible que no sea adecuado desconectar la conexión de cliente existente si hay distintos equipos que están desarrollando aplicaciones AMQP y puede que estén utilizando el mismo ID de cliente. Para resolver este problema puede restringir la toma de control del cliente basándose en el nombre del canal AMQP que se utiliza, la dirección IP del cliente y el ID de usuario del cliente (cuando está habilitada la autenticación SASL).

Utilice los valores de los atributos de gestor de colas **AdoptNewMCA** y **AdoptNewMCACheck** para especificar el nivel necesario de restricción de toma de control, tal como se detalla en la tabla siguiente:

*Tabla 102. Valores de **AdoptNewMCA** y **AdoptNewMCACheck** para restringir la toma de control del cliente*		
AdoptNewMCA	**AdoptNewMCACheck**	**Comprobación de criterios antes de que se permita la toma de control del cliente**
NO o sin definir	No aplicable	Ninguna. La toma de control del cliente está permitida para todas las conexiones de cliente que están autenticadas y pasan todas las reglas de CHLAUTH.
ALL (o un valor distinto de NO)	QM o sin definir	Ninguna. La toma de control del cliente está permitida para todas las conexiones de cliente que están autenticadas y pasan todas las reglas de CHLAUTH.
ALL (o un valor distinto de NO)	NOMBRE	ID de usuario (cuando SASL está habilitado) Nombre de canal
ALL (o un valor distinto de NO)	ADDRESS	ID de usuario (cuando SASL está habilitado) Dirección IP
ALL (o un valor distinto de NO)	TODOS	ID de usuario (cuando SASL está habilitado) Nombre de canal Dirección IP

Los atributos del gestor de colas **AdoptNewMCA** y **AdoptNewMCACheck** forman parte de la configuración del gestor de colas, que se define en la stanza CHANNELS. En IBM MQ para Windows y IBM MQ para sistemas Linux x86-64, modifique la información de configuración utilizando IBM MQ Explorer. En otros sistemas, modifique la información editando el archivo de configuración `qm.ini`. Para obtener información sobre cómo modificar la información de los canales del gestor de colas, consulte Atributos de canales.

Tareas relacionadas

Desarrollo de aplicaciones cliente AMQP

Creación y utilización de canales AMQP

ALW Configuración de JAAS para canales AMQP

Los módulos personalizados de Java Authentication and Authorization Service (JAAS) se pueden utilizar para autenticar las credenciales de nombre de usuario y contraseña que un cliente AMQP pasa a un canal AMQP cuando se conecta.

Acerca de esta tarea

Es posible que desee utilizar un módulo JAAS personalizado si ya utiliza módulos JAAS para la autenticación en otros sistemas basados en Javay desea reutilizar estos módulos para autenticar conexiones AMQP con MQ. De forma alternativa, es posible que desee escribir un módulo JAAS personalizado si las características de autenticación creadas en MQ no dan soporte al mecanismo de autenticación que desea utilizar.

La configuración de los módulos JAAS para los canales AMQP se realiza a un nivel de gestor de colas. Esto significa que, si configura un módulo JAAS para autenticar las conexiones AMQP con el gestor de colas,

el módulo se aplicará a todos los canales AMQP. El nombre del canal que ha invocado el módulo JAAS se pasa al módulo, lo que le permite codificar un registro de JAAS diferente en el comportamiento para diferentes canales.

Otra información también se pasa al módulo JAAS:

- El ID de cliente del cliente AMQP que está intentando autenticarse.
- La dirección de red del cliente AMQP.
- El nombre del canal que ha invocado el módulo JAAS.

Procedimiento

Puede configurar un módulo de configuración JAAS para los canales AMQP completando los pasos siguientes:

1. Defina un archivo `jaas.config` que contenga una o más stanzas de configuración de módulo JAAS. La stanza debe especificar el nombre completo de la clase Java que implementa la interfaz JAAS `javax.security.auth.spi.LoginModule` .
   - Un archivo `jaas.config` predeterminado se suministra con el producto y se encuentra en *QM_data_directory*/amqp/jaas.config.
   - Ya se ha definido una stanza preconfigurada denominada `MQXRConfig` en el archivo `jaas.config` predeterminado.
2. Especifique el nombre de la stanza que se va a utilizar para los canales AMQP.
   - Linux AIX Añada una propiedad al archivo `amqp_unix.properties`.
   - Windows Añada una propiedad al archivo `amqp_win.properties`.

   La propiedad tiene el formato siguiente:

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   Por ejemplo:

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```
3. Configure el entorno del gestor de colas para que incluya la clase del módulo personalizado. El servicio AMQP debe tener acceso a la clase Java configurada en la stanza de configuración JAAS .

   Para ello, añada la vía de acceso a la clase JAAS al archivo `service.env` de MQ. Edite el archivo `service.env` en el directorio de configuración de MQ (*directorio_config_MQ*) o el directorio de configuración del gestor de colas *directorio_config_MQ*) para establecer la variable CLASSPATH en la ubicación de la clase de módulo JAAS.

Qué hacer a continuación

Se proporciona un módulo de inicio de sesión JAAS de ejemplo con el producto en el directorio *mq_installation_directory*/amqp/samples . El módulo de inicio de sesión JAAS de ejemplo autentica todas las conexiones cliente, independientemente del nombre de usuario o de la contraseña con el que se conecta el cliente.

Puede modificar el código fuente del ejemplo y volver a compilarlo para intentar autenticar únicamente a los usuarios específicos con una contraseña determinada. Para configurar el canal AMQP en un sistema UNIX para que utilice el módulo de inicio de sesión JAAS de ejemplo suministrado con el producto:

1. Edite el archivo `/var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties` y establezca la propiedad `com.ibm.mq.MQXR.JAASConfig=MQXRConfig`.
2. Edite el archivo /var/mqm/service.env y establezca la propiedad CLASSPATH=*mq_installation_location*/amqp/samples

El archivo `jaas.config` ya contiene una stanza denominada `MQXRConfig` que especifica la clase de ejemplo `samples.JAASLoginModule` como clase de módulo de inicio de sesión. No es necesario ningún cambio para `jaas.config` antes de probar el módulo de ejemplo.

Tareas relacionadas

Desarrollo de aplicaciones cliente AMQP

Creación y utilización de canales AMQP

Advanced Message Security

Advanced Message Security (AMS) es un componente de IBM MQ que proporciona un alto nivel de protección para los datos confidenciales que fluyen a través de la red IBM MQ, aunque no afecta a las aplicaciones finales.

Visión general de Advanced Message Security

Las aplicaciones de IBM MQ pueden utilizar Advanced Message Security para enviar datos sensibles, tales como transacciones financieras de alto valor e información personal, con niveles diferentes de protección utilizando un modelo de criptografía de clave pública.

Conceptos relacionados

"Intercepción del agente de canal de mensajes (MCA) y AMS" en la página 661

La intercepción MCA permite a un gestor de colas que se ejecuta bajo IBM MQ habilitar de forma selectiva políticas que se van a aplicar para canales de conexión de servidor.

Referencia relacionada

Códigos de retorno de GSKit utilizados en mensajes de AMS

Funciones y características de Advanced Message Security

Advanced Message Security amplía los servicios de seguridad de IBM MQ para proporcionar funciones de firma y cifrado de los datos a nivel de mensaje. Los servicios ampliados garantizan que los datos de los mensajes no se han modificado entre el momento en que se colocaron originalmente en una cola y cuando se recuperaron. Además, AMS verifica que el emisor de los datos de un mensaje está autorizado para colocar mensajes firmados en una cola de destino.

AMS proporciona las siguientes funciones:

- Protege las transacciones sensibles o de alto valor procesadas por IBM MQ.
- Detecta y elimina mensajes no autorizados antes de que sean procesados por una aplicación receptora.
- Verifica que los mensajes no se han modificado mientras estaban en tránsito entre una cola y otra.
- Protege los datos no sólo mientras circulan por la red, sino también cuando se colocan en una cola.
- Protege las aplicaciones propietarias y escritas por el cliente existentes para IBM MQ.
- z/OS A partir de IBM MQ 9.1.3, IBM MQ for z/OS proporciona la posibilidad de eliminar y añadir, de forma opcional, la protección de AMS en los mensajes que fluyen a través de la red, respectivamente. Esto se conoce como *Intercepción del agente de canal de mensajes (MCA) de servidor a servidor*..
- ALW A partir de IBM MQ 9.1.4 y IBM MQ 9.1.0 Fix Pack 4, se añade una comprobación al código de biblioteca IBM MQ que se ejecuta en el programa de aplicación del cliente. La comprobación se ejecuta al principio de su inicialización para leer el valor de la variable de entorno *AMQ_AMS_FIPS_OFF* y, si se establece en cualquier valor, el código IBM Global Security Kit (GSKit) se ejecuta en modalidad no FIPS en dicha aplicación.

Calidades de protección disponibles con AMS

Existen tres calidades de protección para Advanced Message Security, `Integrity`, `Privacy`y `Confidentiality`.

La protección `Integrity` se ofrece mediante la firma digital, que proporciona una garantía sobre quién ha creado el mensaje y que dicho mensaje no se ha modificado ni se ha manipulado indebidamente.

La protección `Privacy` se proporciona mediante una combinación de firma digital y cifrado. El cifrado asegura que los datos del mensaje solo pueda visualizarlos el destinatario o destinatarios deseados. Aunque existan destinatarios no autorizados que obtengan una copia de los datos del mensaje cifrados, no podrán ver los datos reales del mensaje.

La protección de `Confidentiality` se proporciona mediante el cifrado sólo con la reutilización de claves opcional.

Efecto sobre el rendimiento

AMS utiliza una combinación de rutinas criptográficas simétricas y asimétricas para proporcionar la firma digital y el cifrado. Debido a que las operaciones de clave simétrica son muy rápidas en comparación con las operaciones de clave asimétrica, que son intensivas para la CPU, esto puede tener un impacto significativo en los costes de protección de un gran número de mensajes con AMS.

Rutinas criptográficas asimétricas

Por ejemplo, al transferir un mensaje firmado, el código hash del mensaje se firma utilizando una operación de clave asimétrica.

Al obtener un mensaje firmado, se utiliza una operación de clave asimétrica más para verificar el código hash firmado.

Por lo tanto, se necesita un mínimo de dos operaciones de clave asimétrica por mensaje para firmar y verificar los datos del mensaje.

Rutinas criptográficas asimétricas y simétricas

Al transferir un mensaje cifrado, se genera una clave simétrica y luego se cifra utilizando una operación de clave asimétrica para cada destinatario deseado del mensaje.

Los datos del mensaje se cifran a continuación con la clave simétrica. Al obtener el mensaje cifrado, los destinatarios deseados deben utilizar una operación de clave asimétrica para descubrir la clave simétrica que se utiliza para el mensaje.

Por lo tanto, las tres calidades de protección contienen diversos elementos de las operaciones de clave asimétricas que ocupan muchos recursos de CPU, lo cual afectará de forma significativa a la tasa máxima de transferencia de mensajes que se puede obtener en las aplicaciones que transfieren y obtienen mensajes.

Las políticas de `Confidentiality` permiten, no obstante, la reutilización de la clave simétrica en una secuencia de mensajes. Se pueden realizar ahorros de costes de CPU significativos con las políticas de `Confidentiality` a través de la reutilización de claves simétricas. Esta modalidad de operación continúa utilizando el formato PKCS#7 para compartir una clave de cifrado simétrica. No obstante, no hay ninguna firma digital, lo que elimina algunas de las operaciones por clave asimétrica de mensaje. La clave simétrica se debe continuar cifrando con las operaciones de claves asimétricas para cada destinatario, pero la clave simétrica se puede reutilizar opcionalmente en varios mensajes destinados a los mismos destinatarios. Si la política permite la reutilización de claves, solo el primer mensaje requiere operaciones de claves asimétricas. Los mensajes siguientes solo necesitan utilizar operaciones de claves simétricas.

Reutilización de claves

Con las políticas de `Confidentiality` , puede utilizar el enfoque de reutilización de claves simétricas para reducir significativamente los costes implicados en el cifrado de un número de mensajes que se colocan en la misma cola y están pensados para el mismo destinatario o destinatarios.

Por ejemplo, al transferir 10 mensajes cifrados al mismo conjunto de destinatarios, se genera una clave simétrica y, a continuación, se cifra para el primer mensaje, utilizando una operación de clave asimétrica para cada destinatario del mensaje.

En función de los límites controlados por la política, la clave simétrica cifrada se puede reutilizar en los mensajes posteriores que estén dirigidos a los mismos destinatarios. Para permitir que los mensajes posteriores reutilicen la clave simétrica, la aplicación debe mantener la cola abierta después de colocar

un mensaje en la cola. Las operaciones MQPUT1 no pueden reutilizar la clave simétrica. Una aplicación que obtiene mensajes cifrados puede aplicar la misma optimización, con lo cual la aplicación puede detectar cuándo no se ha modificado una clave simétrica y evitar el gasto de recuperar la clave simétrica.

En este ejemplo, el 90% de las operaciones de clave asimétrica se pueden evitar en las aplicaciones de transferencia y de obtención mediante la reutilización de la misma clave.

Para obtener más información sobre cómo realizar la reutilización de clave, consulte:

- Mandato MQSC SET POLICY
- Mandato de control setmqspl
- IBM i Mandato de IBM i SETMQMSPL

Conceptos esenciales de AMS

Conozca los conceptos esenciales de Advanced Message Security para comprender cómo trabaja la herramienta y cómo utilizarla de forma efectiva.

Infraestructura de claves públicas y Advanced Message Security

Una infraestructura de claves públicas (PKI) es un sistema de recursos, políticas y servicios que permiten utilizar la criptografía de clave pública para lograr una comunicación segura.

No existe un estándar individual que defina los componentes de una infraestructura de claves públicas (PKI), pero normalmente una PKI utiliza certificados de clave pública y comprende entidades emisoras de certificados (CA) y otras entidades de registro (RA) que proporcionan los servicios siguientes:

- Emisión de certificados digitales
- Validación de certificados digitales
- Revocación de certificados digitales
- Distribución de certificados

La identidad de los usuarios y las aplicaciones está representada por el campo **nombre distinguido (DN)** en un certificado asociado con mensajes firmados o cifrados. Advanced Message Security utiliza esta identidad para representar un usuario o una aplicación. Para autenticar esta identidad, el usuario o aplicación debe tener acceso al almacén de claves donde se almacenan el certificado y la clave privada asociada. Cada certificado está representado por una etiqueta en el almacén de claves.

Conceptos relacionados

"Utilización de almacenes de claves y certificados con AMS" en la página 655
Para proporcionar protección de cifrado transparente para las aplicaciones de IBM MQ, Advanced Message Security utiliza el archivo de almacén de claves, donde se almacenan certificados de clave pública y una clave privada. En z/OS, se utiliza un conjunto de claves SAF en lugar de un archivo de almacén de claves.

Certificados digitales en AMS

Advanced Message Security asocia usuarios y aplicaciones con certificados digitales X.509 estándar. Normalmente los certificados X.509 están firmados por una entidad emisora de certificados fiable y supone la utilización de claves privadas y públicas para el cifrado y descifrado.

Los certificados digitales proporcionan protección frente a la suplantación de identidad mediante la asociación de una clave pública con su propietario, ya sea un individuo, un gestor de colas o alguna otra entidad. Los certificados digitales también se conocen como certificados de clave pública, pues garantizan la propiedad de una clave pública cuando se utiliza un sistema de claves asimétricas.
Este sistema requiere crear clave pública y una clave privada para una aplicación. Los datos cifrados mediante la clave pública sólo se pueden descifrar mediante la clave privada correspondiente, mientras que los datos cifrados mediante la clave privada sólo se pueden descifrar mediante la clave pública correspondiente. La clave privada se almacena en un archivo de base de datos de claves protegido por contraseña. Sólo el propietario tiene acceso a la clave privada que se utiliza para descifrar los mensajes cifrados mediante la clave pública correspondiente.

Si las claves públicas las envía directamente su propietario a otra entidad, existe el riesgo de que el mensaje pueda ser interceptado y de que la clave pública sea sustituida por otra. Esto se conoce como ataque de interceptor. La solución es intercambiar claves públicas a través de un agente fiable, con lo que el usuario tiene una mayor garantía de que la clave pública pertenece a la entidad con la que se está comunicando. En lugar de enviar la clave pública directamente, el usuario solicita a un agente fiable que la incorpore a un certificado digital. El agente fiable que emite certificados digitales se denomina entidad emisora de certificados.

Para obtener más información sobre los certificados digitales, consulte ¿Qué es un certificado digital?

Un certificado digital contiene la clave pública de una entidad y declara que la clave pública pertenece a esa entidad:

- cuando un certificado es para una entidad individual, se denomina *certificado personal* o *certificado de usuario*.
- cuando un certificado es para una entidad emisora de certificados, el certificado se denomina *certificado de CA* o *certificado de firmante*.

Nota: Advanced Message Security da soporte a los certificados autofirmados en las aplicaciones Java y nativas

Conceptos relacionados

"Criptografía" en la página 11
El cifrado es el proceso de convertir texto legible, denominado *texto plano*, en un formato ilegible, denominado *texto cifrado*.

Multi *Gestor de autorizaciones sobre objetos y AMS*

En Multiplatforms, el gestor de autorizaciones sobre objetos (OAM) es el componente de servicio de autorización que se suministra con los productos IBM MQ.

El acceso a las entidades de Advanced Message Security se controla mediante grupos de usuarios de IBM MQ y el OAM. Los administradores pueden utilizar la interfaz de línea de mandatos para otorgar o revocar autorizaciones según sea necesario. Grupos de usuarios diferentes pueden tener clases diferentes de autorización de acceso para unos mismos objetos. Por ejemplo, un grupo puede realizar operaciones PUT y GET para una cola determinada, mientras que otro grupo puede tener permiso sólo para examinar la cola. De la misma manera, algunos grupos pueden tener autorización GET y PUT para una cola, pero no pueden modificar ni suprimir la cola.

Mediante el OAM, puede controlar lo siguiente:

- Acceso a objetos de Advanced Message Security a través de la interfaz de cola de mensajes (MQI). Cuando un programa de aplicación intenta acceder a objetos, el OAM comprueba si el perfil de usuario que realiza la solicitud tiene autorización para la operación solicitada. Esto significa que las colas y los mensajes de las colas se pueden proteger contra el acceso no autorizado.
- El permiso para utilizar mandatos PCF y MQSC.

Conceptos relacionados

Gestor de autorizaciones sobre objetos

Descripción general de la interfaz de cola de mensajes (Message Queue Interface, MQI)

Tecnología soportada por Advanced Message Security

Advanced Message Security depende de varios componentes de tecnología para proporcionar una infraestructura de seguridad.

Advanced Message Security es compatible con las siguientes interfaces de programación de aplicaciones (API) de IBM MQ:

- Interfaz de cola de mensajes (MQI)
- IBM MQ Java Message Service (JMS) 1.0.2 y 1.1.
- Clases base de IBM MQ para Java

- Clases de IBM MQ para .Net en modalidad no gestionada

Nota: Advanced Message Security es compatible con las entidades emisoras de certificados que cumplen la especificación X.509.

Limitaciones conocidas de AMS

Existe un número de opciones de IBM MQ que no están soportadas o que tienen limitaciones para Advanced Message Security.

- Las siguientes opciones de IBM MQ no están soportadas o tienen limitaciones:

  Publicación/suscripción
  Una de las principales ventajas de un modelo de mensajería publicación/suscripción respecto de un modelo punto a punto es que las aplicaciones emisora y receptora no necesitan saber nada la una de la otra para que los datos que se envíen y reciban. Esta ventaja se pierde con el uso de políticas Advanced Message Security en las que hay que definir los destinatarios o firmantes autorizados. Es posible que una aplicación publique en un tema a través de una definición de cola de alias que esté protegida por una política, también es posible que una aplicación suscriptora obtenga mensajes de una cola protegida por política. No es posible asignar una política directamente a una cadena de tema, las políticas solo pueden asignarse a definiciones de cola.

  Conversión de datos de canal
  La carga útil protegida de un mensaje protegido de Advanced Message Security se transmite en formato binario, lo que garantiza que la conversión de datos en un canal entre aplicaciones no invalide el resumen de mensaje (digest). Las aplicaciones que recuperan mensajes de una cola protegida por política tienen que solicitar una conversión de datos, la conversión de la carga útil protegida se intentará una vez verificados y desprotegidos correctamente los mensajes.

  Listas de distribución
  Las políticas de Advanced Message Security pueden usarse al proteger aplicaciones colocando mensajes en listas de distribución, siempre que cada cola de destino de la lista tenga definida una política idéntica. Si se identifican políticas incoherentes cuando una aplicación abre una lista de distribución, la operación de apertura (open) fallará y se devolverá un error de seguridad a la aplicación.

  Segmentación de mensajes de aplicación
  El tamaño de los mensajes protegidos por política aumentará y no es posible que las aplicaciones especifiquen con precisión los límites de segmento de un mensaje.

  Aplicaciones que utilizan IBM MQ classes for .NET en un nodo gestionado (conexiones de cliente)
  Las aplicaciones que usan IBM MQ classes for .NET en modo gestionado (conexiones de cliente) no están soportadas.

  Nota: La intercepción MCA se puede utilizar para permitir a clientes no soportados utilizar AMS.

  El cliente de servicio de mensajería para aplicaciones .NET (XMS) en una modalidad gestionada
  Los clientes de servicio de mensajería de aplicaciones .NET (XMS) en modo gestionado no están soportados.

  Nota: La intercepción MCA puede utilizarse para permitir que los clientes soportados usen AMS.

  Colas IBM MQ procesadas por el puente IMS
  Las colas IBM MQ procesadas por el puente IMS no están soportadas.

  Nota: AMS está soportado en colas Puente CICS. Debe utilizar el mismo ID de usuario para MQPUT (cifrar) y MQGET (descifrar) en colas Puente CICS.

  Colocación en espera de método de obtención
  La colocación en espera del método de obtención no está soportada para las aplicaciones de obtención respecto a colas que tienen políticas AMS definidas para ellas.

  z/OS **Intercepción de agente de canal de mensajes de servidor a servidor**
  A partir de IBM MQ for z/OS 9.1.3, la intercepción de MCA de servidor a servidor sólo está soportada para los tipos de canal emisor, servidor, receptor y peticionario.

- Los usuarios deben evitar colocar más de un certificado con el mismo nombre distinguido en un único archivo de almacén de claves, porque la selección de qué certificado usar al proteger un mensaje no está definida.
- AMS no está soportado en JMS si la propiedad **WMQ_PROVIDER_VERSION** se establece en 6.
- El interceptor AMS no está soportado para canales AMQP o 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.

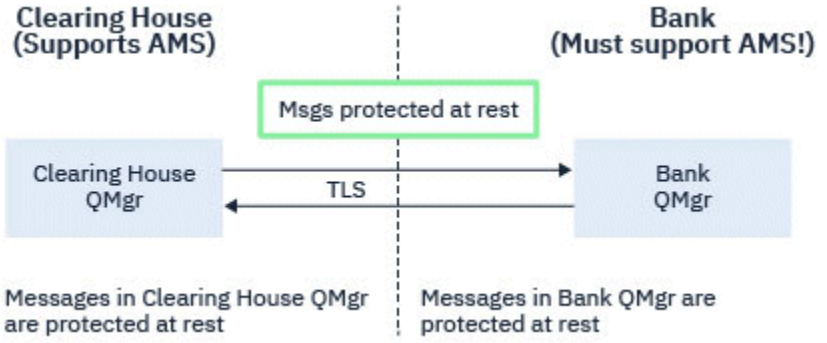


Figure 32. Usage of AMS without AMS interception

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.

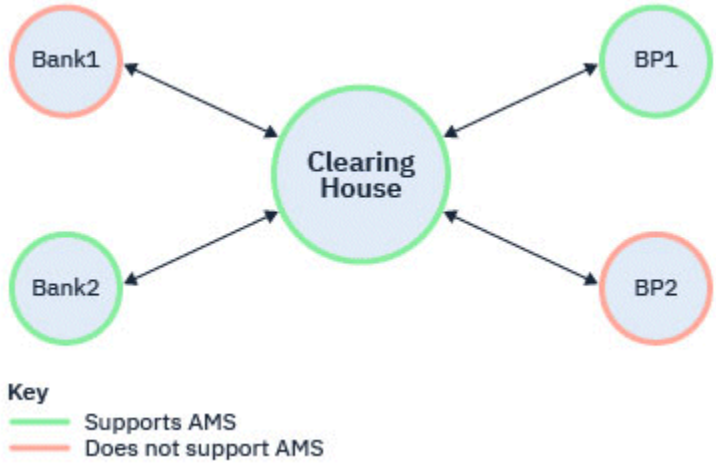


Figure 33. Some partners support AMS and some do not

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

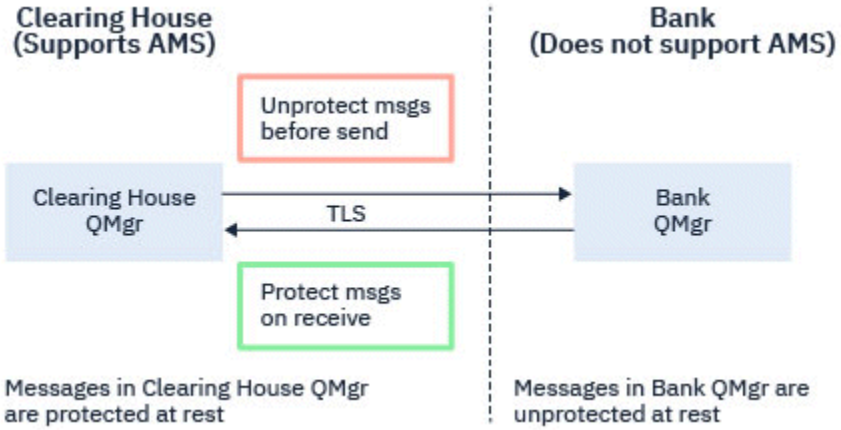


Figure 34. Message flow between business partners

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
: Pasa por los mensajes enviados o recibidos por el agente de canal de mensajes de este canal, sin modificarlos.

  Este valor es válido para canales con un tipo de canal (**CHLTYPE**) de SDR, RVS, RCVR o RQSTR, y es el valor predeterminado.

REMOVE
: Elimine cualquier protección de AMS de los mensajes recuperados de la cola de transmisión por el agente de canal de mensajes y envíe los mensajes al socio.

  Cuando el agente de canal de mensajes obtiene un mensaje de la cola de transmisión, si se ha definido una política de AMS para la cola de transmisión, se aplica para eliminar cualquier protección de AMS del mensaje antes de enviar el mensaje a través del canal. Si no se ha definido una política AMS para la cola de transmisión, el mensaje se enviará tal cual.

  Este valor solo es válido para canales con un tipo de canal de SDR o SVR.

ASPOLICY
: Basándose en la política definida para la cola de destino, aplique la protección de AMS a los mensajes de entrada antes de colocarlos en la cola de destino.

Cuando el agente de canal de mensajes recibe un mensaje de entrada, si se ha definido una política de AMS para la cola de destino, la protección de AMS se aplica al mensaje antes de que el mensaje se coloque en la cola de destino. Si no se ha definido una política de AMS para la cola de destino, el mensaje se coloca en la cola de destino tal como está.

Este valor solo es válido para canales con un tipo de canal de RCVR o 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

Manejo de errores para AMS

IBM MQ Advanced Message Security define una cola de tratamiento de errores para gestionar los mensajes que contienen errores o los mensajes que no se pueden desproteger.

Los mensajes defectuosos se tratan como casos excepcionales. Si un mensaje recibido no cumple los requisitos de seguridad de la cola en la que se encuentra, por ejemplo, si el mensaje está firmado cuando debería estar cifrado, o si fallan el descifrado o l verificación de la firma, el mensaje se envía a la cola de tratamiento de errores. Un mensaje se puede enviar a la cola de tratamiento de errores por las siguientes razones:

- Discrepancia de calidad de protección: existe una discrepancia en la calidad de protección (QOP) entre el mensaje recibido y la definición QOP de la política de seguridad.
- Error de descifrado - el mensaje no puede descrifrarse.
- Error de cabecera PDMQ - no se puede acceder a la cabecera de mensaje de Advanced Message Security (AMS).
- Discrepancia de tamaños - la longitud de un mensaje tras el descifrado es distinta de la esperada.

- Discrepancia de fuerza del algoritmo de cifrado: el algoritmo de cifrado del mensaje no tiene la fuerza necesaria.
- Error desconocido: se ha producido un error inesperado.

AMS utiliza SYSTEM.PROTECTION.ERROR.QUEUE como cola de manejo de errores. Todos los mensajes colocados por IBM MQ AMS en SYSTEM.PROTECTION.ERROR.QUEUE van precedidos de una cabecera MQDLH.

El administrador de IBM MQ también puede definir el SYSTEM.PROTECTION.ERROR.QUEUE como una cola alias que apunta a otra cola.

z/OS En IBM MQ for z/OS, si se está utilizando la intercepción del agente de canal de mensajes (MCA) de servidor a servidor:

- Si por una de las razones anteriormente indicadas, IBM MQ AMS mueve los mensajes de la cola de transmisión a la cola de tratamiento de errores, el MCA emisor simplemente procede a procesar el siguiente mensaje disponible en la cola de transmisión.
- En general, se aplican las reglas de canal existentes para:
  - Colocar mensajes en la cola de mensajes no entregados
  - Acciones realizadas en caso de que fallara la colocación en la cola de mensajes no entregados.

Consulte “Mensajes no entregados para AMS en z/OS” en la página 618 para obtener más información sobre casos específicos.

z/OS *Mensajes no entregados para AMS en z/OS*

Escenarios específicos relacionados con la interceptación del agente de canal de mensajes de servidor a servidor en IBM MQ for z/OS.

En IBM MQ for z/OS, si se está utilizando la intercepción del agente de canal de mensajes (MCA) de servidor a servidor:

- Si, después de haber obtenido un mensaje no protegido, el MCA emisor no puede entregar un mensaje por alguna razón, por ejemplo, porque el mensaje es demasiado grande para el canal, si el atributo de canal emisor USEDLQ se establece en YES, el MCA emisor traslada el mensaje a la cola de mensajes no entregados local (DLQ).

  Si SYSTEM.DEAD.LETTER.QUEUE se está utilizando como la cola de mensajes no entregados local, el mensaje se colocará sin protección.

  Nota: IBM MQ AMS no admite la protección de mensajes puestos en las colas de sistema.

  Si se utiliza una cola de mensajes no entregados con nombre como la cola de mensajes no entregados local, el mensaje se colocará protegido si ha definido una política de IBM MQ AMS con el mismo nombre que la cola de mensajes no entregados nombrada y se colocará desprotegida si no ha definido una política adecuada.
- Si no se puede poner un mensaje en la cola de mensajes no entregados local por algún motivo, si la NPMSPEED del canal se establece en NORMAL o el mensaje es permanente, se restituye el lote actual de mensajes y el canal se pone en estado RETRY (Reintentar). De lo contrario, el mensaje se descarta y el agente de canal de mensajes emisor continúa procesando el siguiente mensaje en la cola de transmisión.
- Dado que las políticas de seguridad no tienen ningún efecto sobre la cola SYSTEM.DEAD.LETTER.QUEUE o las otras colas SYSTEM enumeradas en “Protección de colas del sistema en AMS” en la página 692, si SYSTEM.DEAD.LETTER.QUEUE está en uso, los mensajes puestos en esta cola por los agentes de canal de mensajes se colocan tal cual están. Es decir, si los mensajes se han protegido anteriormente, se colocan protegidos; de lo contrario, se colocan desprotegidos.

  Si el atributo DEADQ del gestor de colas se ha establecido en el nombre de una cola de mensajes no entregados alternativa (no del sistema) y no existe una política de AMS con el mismo nombre, los mensajes colocados en esta cola por los agentes de canal de mensajes se colocan tal cual están. Es

decir, si los mensajes se han protegido anteriormente, se colocan protegidos; de lo contrario, se colocan desprotegidos.

Si el atributo DEADQ del gestor de colas se ha establecido en el nombre de una cola de mensajes no entregados alternativa (no del sistema) y existe una política de AMS con el mismo nombre, los mensajes colocados en esta cola por los agentes de canal de mensajes se colocan tal cual están. Si el mensaje ya se ha protegido, no se vuelve a proteger; esto es para evitar una doble protección. Si no existe una política de servicios de método de acceso con el mismo nombre, los mensajes se colocan tal cual.

- Si hay una política para la cola de mensajes no entregados con la opción de tolerancia en el conjunto de mandatos setmqspl establecida en desactivada, es decir '**-t** O', la colocación en cola de mensajes no entregados falla si el mensaje no está protegido por AMS y, por lo tanto, no tiene una cabecera PDMQ. Esto sucede si el mensaje llega al receptor sin una cabecera PDMQ. Es el putter original del mensaje que no tenía una política para el destino y el receptor no tiene establecido SPLPROT (ASPOLICY).
- Es posible que un MCA no pueda colocar un mensaje en la cola de mensajes no entregados, si la política de AMS definida para la cola de mensajes no entregados no permite el ID de usuario con el que se ejecuta el iniciador de canal para proteger el mensaje.
- Los canales receptores suelen colocar mensajes no entregados en la cola de mensajes no entregados local, mientras que los canales emisores suelen colocar los mensajes que no se pueden procesar por alguna razón, por ejemplo, los mensajes demasiado grandes para la cola, o una cabecera MQXQH incorrecta, etc. en la cola de mensajes no entregados local.
- Los manejadores de cola de mensajes no entregados normalmente solo buscan en la cabecera de cola de mensajes no entregados (DLH) y no en la carga útil de mensaje en sí. Por lo tanto, el hecho de que la carga útil de mensaje pueda estar protegida, no impide que los manejadores determinen por qué se ha colocado el mensaje en la cola de mensajes no entregados.
- Si no se ha definido una cola de mensajes no entregados, el canal:
  - Finaliza de forma anómala (y entra en estado de reintento) si no se puede entregar un mensaje permanente.
  - Descarta un mensaje no entregado no permanente y continúa ejecutándose.

Conceptos relacionados

“Manejo de errores para AMS” en la página 617
IBM MQ Advanced Message Security define una cola de tratamiento de errores para gestionar los mensajes que contienen errores o los mensajes que no se pueden desproteger.

Escenarios de usuario para AMS

Conozca los posibles escenarios para comprender cuáles son los objetivos empresariales que se pueden alcanzar con Advanced Message Security.

Windows *Guía de inicio rápido para AMS en plataformas Windows*

Utilice esta guía para configurar rápidamente Advanced Message Security (AMS) para proporcionar seguridad de mensajes en plataformas Windows . La guía describe cómo crear una base de datos para verificar las identidades de usuario y definir políticas de firma/cifrado para el gestor de colas.

Antes de empezar

Como mínimo, es necesario tener instaladas en el sistema las siguientes características:

- Servidor
- Kit de herramientas de desarrollo (para los programas de ejemplo)
- Advanced Message Security (AMS)

Consulte las características de IBM MQ para sistemas Windows para obtener información detallada.

Para obtener información sobre cómo utilizar el mandato **setmqenv** para inicializar el entorno actual para que el sistema operativo pueda localizar y ejecutar los mandatos IBM MQ adecuados, consulte setmqenv (set IBM MQ environment).

1. Crear un gestor de colas y una cola

Acerca de esta tarea

Todos los ejemplos siguientes utilizan una cola denominada TEST.Q para pasar mensajes entre aplicaciones. Advanced Message Security utiliza interceptores para firmar y cifrar mensajes en el momento en que los mensajes entran en la infraestructura de IBM MQ a través de la interfaz estándar de IBM MQ. La configuración básica se realiza en IBM MQ y se define en los pasos siguientes.

Puede utilizar IBM MQ Explorer para crear el gestor de colas QM_VERIFY_AMS y su cola local denominada TEST.Q utilizando todos los valores predeterminados del asistente, o puede utilizar los mandatos que se encuentran en `C:\Archivos de programa\IBM\MQ\bin`. Recuerde que debe ser miembro del grupo de usuarios `mqm` para ejecutar los siguientes mandatos administrativos.

Procedimiento

1. Crear un gestor de colas

```
crtmqm QM_VERIFY_AMS
```

2. Inicie el gestor de colas

```
strmqm QM_VERIFY_AMS
```

3. Cree una cola denominada TEST.Q especificando el siguiente mandato en **runmqsc** para el gestor de colas QM_VERIFY_AMS

```
DEFINE QLOCAL(TEST.Q)
```

Resultados

Si se completa el procedimiento, el mandato introducido en **runmqsc** mostrará detalles sobre TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Crear y autorizar usuarios

Acerca de esta tarea

En este ejemplo existen dos usuarios: alice, el emisor, y bob, el receptor. Para utilizar la cola de aplicación, estos usuarios deben tener autorización para utilizarla. Asimismo, para utilizar correctamente las políticas de protección que vamos a definir, estos usuarios deben tener acceso a algunas colas del sistema. Para obtener más información sobre el mandato **setmqaut** , consulte **setmqaut**.

Procedimiento

1. Cree los dos usuarios y asegúrese de que HOMEPATH y HOMEDRIVE se hayan establecido para estos usuarios.
2. Autorice a los usuarios para conectarse con el gestor de colas y para trabajar con la cola

```
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. También debe permitir que los dos usuarios examinen la cola de políticas del sistema y coloquen mensajes en la cola de errores.

```
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
```

Atención: IBM MQ optimiza el rendimiento almacenando en memoria caché las políticas para que no tenga que examinar los registros para obtener detalles de política en el SYSTEM.PROTECTION.POLICY.QUEUE en todos los casos.

IBM MQ no almacena en memoria caché todas las políticas disponibles. Si hay un número alto de políticas, IBM MQ almacena en memoria caché un número limitado de políticas. Por lo tanto, si el gestor de colas tiene un número bajo de políticas definidas, no es necesario proporcionar la opción de examinar a SYSTEM.PROTECTION.POLICY.QUEUE.

Sin embargo, debe otorgar autorización para examinar esta cola, en caso de que haya un gran número de políticas definidas, o si está utilizando clientes antiguos. El sistema SYSTEM.PROTECTION.ERROR.QUEUE se utiliza para colocar mensajes de error generados por el código AMS. La autorización de colocación sobre esta cola sólo se comprueba cuando se intenta colocar un mensaje de error en la cola. La autorización de colocación sobre la cola no se comprueba cuando intenta transferir u obtener un mensaje de una cola protegida por AMS.

Resultados

Se crean los usuarios y se les otorgan las autorizaciones necesarias.

Qué hacer a continuación

Para verificar si los pasos se han realizado correctamente, utilice los ejemplos `amqsput` y `amqsget` tal como se describe en la sección “7. Probar la configuración” en la página 624.

3. Crear la base de datos de claves y certificados

Acerca de esta tarea

El interceptor necesita la clave pública de los usuarios de los usuarios emisores para cifrar el mensaje. Por lo tanto, se deben crear la base de datos de claves de identidades de usuario que están correlacionadas con claves públicas y privadas. En el sistema real, donde los usuarios y las aplicaciones están distribuidos en varios sistemas, cada usuario tendría su propio almacén de claves privado. De forma similar, en esta guía, creamos bases de datos de claves para `alice` y bob y compartimos los certificados de usuario entre ellos.

Nota: En esta guía, utilizamos aplicaciones de ejemplo escritas en C que se conectan mediante enlaces locales. Si tiene previsto utilizar aplicaciones Java utilizando enlaces de cliente, debe crear un almacén de claves JKS y certificados utilizando el mandato Java **`keytool`** V 9.4.0 V 9.4.0 o el mandato IBM MQ **`runmqktool`**. Para obtener más información, consulte “Guía de inicio rápido para AMS con clientes Java” en la página 642. Para los demás lenguajes, y para las aplicaciones Javaque utilizan enlaces locales, los pasos de esta guía son correctos.

Procedimiento

1. Cree una nueva base de datos de claves para el usuario `alice`.
   Por ejemplo, emita el mandato siguiente para crear la nueva base de datos de claves:

```
runmqakm -keydb -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -type cms -pw
passw0rd -stash
```

   Nota:

- Utilice una contraseña segura para proteger la base de datos.
- Incluya el parámetro **-stash** para ocultar la contraseña de base de datos de claves cifrada en un archivo.

2. Cree un nuevo certificado autofirmado para identificar el usuario `alice` para utilizarlo en el cifrado. Por ejemplo, emita el mandato siguiente para crear un nuevo certificado autofirmado:

```
runmqakm -cert -create -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -stashed
-label Alice_Cert -dn "CN=alice, O=IBM, C=GB"
```

Nota:

- A los efectos de esta guía, utilizamos un certificado autofirmado que se puede crear sin utilizar una entidad emisora de certificados. Para los sistemas de producción, es aconsejable utilizar certificados firmados por una entidad emisora de certificados.
- El parámetro **-label** especifica el nombre del certificado, que los interceptores buscarán para recibir la información necesaria.
- El parámetro **-dn** especifica los detalles del nombre distinguido (DN) del certificado. El nombre distinguido debe ser exclusivo para cada usuario.

3. Repita los pasos “1” en la página 621 y “2” en la página 622 para el usuario bob.

Resultados

Los dos usuarios `alice` y bob tienen cada uno un certificado autofirmado.

4. Crear keystore.conf

Acerca de esta tarea

Debe apuntar los interceptores de Advanced Message Security al directorio donde residen las bases de datos y certificados. Esto se realiza a través del archivo `keystore.conf` , que contiene esa información en formato de texto sin formato. Cada usuario debe tener un archivo `keystore.conf` independiente en la carpeta `.mqs`. Este paso debe realizarse tanto para `alice` como para bob.

El contenido de `keystore.conf` debe tener este formato:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

Ejemplo

Para este caso de ejemplo, el contenido de `keystore.conf` será el siguiente:

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

Nota:

- La vía de acceso del archivo de almacén de claves se debe especificar sin ninguna extensión de archivo.
- La etiqueta del certificado puede incluir espacios, por lo que "Alice_Cert" y "Alice_Cert" (con un espacio al final) por ejemplo, se reconocen como etiquetas de dos certificados diferentes. Sin embargo, para evitar confusiones, es mejor no utilizar espacios en el nombre de la etiqueta.
- Existen los siguientes formatos de almacén de claves: CMS (Cryptographic Message Syntax), JKS (Java Keystore) y JCEKS (Java Cryptographic Extension Keystore). Para obtener más información, consulte “Estructura del archivo de configuración del almacén de claves (keystore.conf) para AMS” en la página 656.
- *%HOMEDRIVE%\%HOMEPATH%*\.mqs\keystore.conf (p.ej., C:\Documents and Settings\alice\.mqs\keystore.conf) es la ubicación predeterminada donde Advanced Message Security busca el archivo `keystore.conf`. Para obtener información sobre cómo utilizar una ubicación no

predeterminada para `keystore.conf`, consulte “Utilización de almacenes de claves y certificados con AMS” en la página 655.

- Para crear el directorio `.mqs` debe usar el indicador de mandatos.

5. Compartir certificados

Acerca de esta tarea

Comparta los certificados entre las dos bases de datos para que cada usuario pueda identificar correctamente al otro. Esto se realiza extrayendo el certificado público de cada usuario en un archivo, que después se añade a la base de datos de claves del otro usuario.

Nota: Tenga cuidado y utilice la opción *extraer* y no la opción *exportar*. *Extraer* obtiene la clave pública del usuario, mientras que *exportar* obtiene ambas claves, la pública y la privada. El uso de *exportar* por error comprometería por completo la aplicación, pasando su clave privada.

Procedimiento

1. Extraiga el certificado que identifica a `alice` a un archivo externo.

```
runmqakm -cert -extract -db "C:\Documents and Settings\alice\AMS\alicekey.kdb" -pw passw0rd
-label Alice_Cert -target alice_public.arm
```

2. Añada el certificado al almacén de clavos `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 los pasos 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

Los dos usuarios `alice` y bob pueden identificar ahora correctamente al otro mediante la creación y compartición de certificados autofirmados.

Qué hacer a continuación

Para verificar que un certificado está en el almacén de claves, examínelo utilizando la GUI o ejecute los siguientes mandatos que imprimen los detalles:

```
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. Definir la política de cola

Acerca de esta tarea

Después de crear el gestor de colas y preparar interceptores para interceptar mensajes y acceder a claves de cifrado, podemos comenzar a definir políticas de protección en `QM_VERIFY_AMS` mediante el

mandato `setmqspl`. Consulte setmqspl para obtener más información sobre este mandato. Cada nombre de política debe ser el mismo que el nombre de cola al que se debe aplicar.

Ejemplo

Este es un ejemplo de una política definida para la cola `TEST.Q`. En el ejemplo, los mensajes se firman con el algoritmo Deprecated SHA1 y se cifran con el algoritmo AES256. `alice` es el único emisor válido y bob es el único receptor de los mensajes en esta cola:

```
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: Los DN coinciden exactamente con los especificados en el certificado del usuario respectivo de la base de datos de claves.

Qué hacer a continuación

Para verificar la política que ha definido, emita el mandato siguiente:

```
dspmqspl -m QM_VERIFY_AMS
```

Para imprimir los detalles de la política como un conjunto de mandatos `setmqspl`, utilice el distintivo `-export`. Esto permite almacenar políticas ya definidas:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Probar la configuración

Acerca de esta tarea

Puede ejecutar programas diferentes bajo usuarios diferentes para verificar si la aplicación se ha configurado debidamente.

Procedimiento

1. Cambie el usuario de modo que se ejecute como usuario `alice`

   Pulse con el botón derecho del ratón en `cmd.exe` y seleccione **Ejecutar como...**. Cuando se le solicite, inicie una sesión como el usuario `alice`.
2. Como usuario `alice`, coloque un mensaje utilizando una aplicación de ejemplo:

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. Escriba el texto del mensaje y pulse Intro.
4. Cambie el usuario de modo que se ejecute como usuario bob

   Pulse con el botón derecho del ratón en `cmd.exe` y seleccione **Ejecutar como...** para abrir otra ventana. Cuando se le solicite, inicie una sesión como el usuario bob.
5. Como usuario bob, obtenga un mensaje utilizando una aplicación de ejemplo:

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

Resultados

Si la aplicación se ha configurado debidamente para ambos usuarios, el mensaje del usuario `alice` se visualiza cuando bob ejecuta la aplicación de obtención.

8. Probar el cifrado

Acerca de esta tarea

Para verificar que el cifrado se realiza según lo esperado, cree una cola de alias que haga referencia a la cola original TEST.Q. Esta cola de alias no tendrá ninguna política de seguridad, por lo que ningún usuario tendrá la información para descifrar el mensaje y, por lo tanto, se mostrarán los datos cifrados.

Procedimiento

1. Utilizando el mandato **runmqsc** en el gestor de colas QM_VERIFY_AMS, cree una cola de alias.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Otorgue a bob el acceso para examinar desde la cola de alias

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. Como usuario alice, coloque otro mensaje utilizando una aplicación de ejemplo al igual que antes:

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. Como usuario bob, examine el mensaje utilizando una aplicación de ejemplo utilizando la cola de alias esta vez:

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. Como usuario bob, obtenga el mensaje utilizando una aplicación de ejemplo desde la cola local:

```
amqsget TEST.Q QM_VERIFY_AMS
```

Resultados

La salida de la aplicación amqsbcg muestra los datos cifrados que hay en la cola, lo que demuestra que el mensaje se ha cifrado.

Linux AIX *Guía de inicio rápido para AMS en AIX and Linux*

Use esta guía para configurar rápidamente Advanced Message Security para proporcionar seguridad de mensajes en AIX and Linux. La guía describe cómo crear una base de datos para verificar las identidades de usuario y definir políticas de firma/cifrado para el gestor de colas.

Antes de empezar

Como mínimo, es necesario tener instalados en el sistema los siguientes componentes:

- Tiempo de ejecución
- Servidor
- Programas de ejemplo
- IBM Global Security Kit (GSKit)
- Advanced Message Security

Consulte los temas siguientes para ver los nombres de componentes en cada plataforma específica:

- Linux Componentes de IBM MQ para sistemas Linux
- AIX Componentes de IBM MQ para sistemas AIX

1. Crear un gestor de colas y una cola

Acerca de esta tarea

Todos los ejemplos siguientes utilizan una cola denominada TEST.Q para pasar mensajes entre aplicaciones. Advanced Message Security utiliza interceptores para firmar y cifrar mensajes en el momento en que los mensajes entran en la infraestructura de IBM MQ a través de la interfaz estándar de IBM MQ. La configuración básica se realiza en IBM MQ y se define en los pasos siguientes.

Puede utilizar IBM MQ Explorer para crear el gestor de colas QM_VERIFY_AMS y su cola local denominada TEST.Q utilizando todos los valores predeterminados del asistente, o puede utilizar los mandatos que se encuentran en *MQ_INSTALLATION_PATH*/bin. Recuerde que debe ser miembro del grupo de usuarios mqm para ejecutar los siguientes mandatos administrativos.

Procedimiento

1. Crear un gestor de colas

```
crtmqm QM_VERIFY_AMS
```

2. Inicie el gestor de colas

```
strmqm QM_VERIFY_AMS
```

3. Cree una cola denominada TEST.Q especificando el siguiente mandato en **runmqsc** para el gestor de colas QM_VERIFY_AMS

```
DEFINE QLOCAL(TEST.Q)
```

Resultados

Si el procedimiento se ha ejecutado correctamente, el siguiente mandato especificado en **runmqsc** mostrará detalles sobre TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Crear y autorizar usuarios

Acerca de esta tarea

En este ejemplo existen dos usuarios: alice, el emisor, y bob, el receptor. Para utilizar la cola de aplicación, estos usuarios deben tener autorización para utilizarla. Asimismo, para utilizar correctamente las políticas de protección que vamos a definir, estos usuarios deben tener acceso a algunas colas del sistema. Para obtener más información sobre el mandato **setmqaut** , consulte **setmqaut**.

Procedimiento

1. Cree los dos usuarios.

```
useradd alice
```

```
useradd bob
```

2. Autorice a los usuarios para conectarse con el gestor de colas y para trabajar con la cola

```
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. También debe permitir que los dos usuarios examinen la cola de políticas del sistema y coloquen mensajes en la cola de errores.

```
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
```

Atención: IBM MQ optimiza el rendimiento almacenando en memoria caché las políticas para que no tenga que examinar los registros para obtener detalles de política en el SYSTEM.PROTECTION.POLICY.QUEUE en todos los casos.

IBM MQ no almacena en memoria caché todas las políticas disponibles. Si hay un número alto de políticas, IBM MQ almacena en memoria caché un número limitado de políticas. Por lo tanto, si el gestor de colas tiene un número bajo de políticas definidas, no es necesario proporcionar la opción de examinar a SYSTEM.PROTECTION.POLICY.QUEUE.

Sin embargo, debe otorgar autorización para examinar esta cola, en caso de que haya un gran número de políticas definidas, o si está utilizando clientes antiguos. El sistema SYSTEM.PROTECTION.ERROR.QUEUE se utiliza para colocar mensajes de error generados por el código AMS. La autorización de colocación sobre esta cola sólo se comprueba cuando se intenta colocar un mensaje de error en la cola. La autorización de colocación sobre la cola no se comprueba cuando intenta transferir u obtener un mensaje de una cola protegida por AMS.

Resultados

Se crean los grupos de usuarios y se les otorgan las autorizaciones necesarias. De este forma, los usuarios que se asignen a esos grupos también tendrán permisos para conectarse al gestor de colas y realizar operaciones put y get con la cola.

Qué hacer a continuación

Para verificar si los pasos se han realizado correctamente, utilice los ejemplos `amqsput` y `amqsget` tal como se describe en la sección “8. Probar el cifrado” en la página 631.

3. Crear la base de datos de claves y certificados

Acerca de esta tarea

Para cifrar el mensaje, el interceptor necesita la clave privada del usuario emisor y las claves públicas del destinatario o los destinatarios. Por lo tanto, se deben crear la base de datos de claves de identidades de usuario que están correlacionadas con claves públicas y privadas. En el sistema real, donde los usuarios y las aplicaciones están distribuidos en varios sistemas, cada usuario tendría su propio almacén de claves privado. De forma similar, en esta guía, creamos bases de datos de claves para `alice` y bob y compartimos los certificados de usuario entre ellos.

Nota: En esta guía, utilizamos aplicaciones de ejemplo escritas en C que se conectan mediante enlaces locales. Si tiene previsto utilizar aplicaciones Java utilizando enlaces de cliente, debe crear un almacén de claves de JKS y certificados mediante el mandato **`keytool`**, que forma parte del JRE (consulte “Guía de inicio rápido para AMS con clientes Java” en la página 642 para obtener más detalles). Para los demás lenguajes, y para las aplicaciones Javaque utilizan enlaces locales, los pasos de esta guía son correctos.

Procedimiento

1. Cree una nueva base de datos de claves para el usuario `alice`

```
mkdir /home/alice/.mqs -p
```

```
runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
```

Nota:

- Se recomienda utilizar una contraseña fuerte para proteger la base de datos.
- El parámetro **`stash`** almacena la contraseña en el archivo `key.sth`, que los interceptores pueden utilizar para abrir la base de datos.

2. Asegúrese de que la base de datos de claves sea legible.

```
chmod +r /home/alice/.mqs/alicekey.kdb
```

3. Cree un certificado que identifique al usuario `alice` para utilizarlo en el cifrado

```
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:

- A los efectos de esta guía, utilizamos un certificado autofirmado que se puede crear sin utilizar una entidad emisora de certificados. Para los sistemas de producción, se recomienda no utilizar certificados autofirmados, sino certificados firmados por una entidad emisora de certificados.
- El parámetro **`label`** especifica el nombre para el certificado, que los interceptores buscarán para recibir información necesaria.
- El parámetro **DN** especifica los detalles del **Nombre distinguido** (DN), que debe ser exclusivo para cada usuario.

4. Ahora que hemos creado la base de datos de claves, debemos establecer su propiedad y comprobar que sea ilegible para los demás usuarios.

```
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 los pasos del 1 al 4 para el usuario bob

Resultados

Los dos usuarios `alice` y bob tienen cada uno un certificado autofirmado.

4. Crear keystore.conf

Acerca de esta tarea

Debe apuntar los interceptores de Advanced Message Security al directorio donde residen las bases de datos y certificados. Esto se realiza a través del archivo `keystore.conf` , que contiene esa información en formato de texto sin formato. Cada usuario debe tener un archivo `keystore.conf` independiente en la carpeta `.mqs`. Este paso debe realizarse tanto para `alice` como para bob.

El contenido de `keystore.conf` debe tener este formato:

```
cms.keystore = dir/keystore_file
```

```
cms.certificate = certificate_label
```

Ejemplo

Para este caso de ejemplo, el contenido de `keystore.conf` será el siguiente:

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

Nota:

- La vía de acceso del archivo de almacén de claves se debe especificar sin ninguna extensión de archivo.
- Existen los siguientes formatos de almacén de claves: CMS (Cryptographic Message Syntax), JKS (Java Keystore) y JCEKS (Java Cryptographic Extension Keystore). Para obtener más información, consulte "Estructura del archivo de configuración del almacén de claves (keystore.conf) para AMS" en la página 656.
- `HOME/.mqs/keystore.conf` es la ubicación predeterminada donde Advanced Message Security busca el archivo `keystore.conf`. Para obtener información sobre cómo utilizar una ubicación no predeterminada para `keystore.conf`, consulte "Utilización de almacenes de claves y certificados con AMS" en la página 655.

5. Compartir certificados

Acerca de esta tarea

Comparta los certificados entre las dos bases de datos para que cada usuario pueda identificar correctamente al otro. Esto se realiza extrayendo el certificado público de cada usuario en un archivo, que después se añade a la base de datos de claves del otro usuario.

Nota: Tenga cuidado y utilice la opción *extraer* y no la opción *exportar*. *Extraer* obtiene la clave pública del usuario, mientras que *exportar* obtiene ambas claves, la pública y la privada. El uso de *exportar* por error comprometería por completo la aplicación, pasando su clave privada.

Procedimiento

1. Extraiga el certificado que identifica a `alice` a un archivo externo.

   ```
   runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
   -target alice_public.arm
   ```

2. Añada el certificado al almacén de clavos `bob's`:

   ```
   runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file
   alice_public.arm
   ```

3. Repita el paso para bob:

   ```
   runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
   bob_public.arm
   ```

4. Añada el certificado para bob al almacén de claves `alice's`:

   ```
   runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
   bob_public.arm
   ```

Resultados

Los dos usuarios `alice` y bob pueden identificar ahora correctamente al otro mediante la creación y compartición de certificados autofirmados.

Qué hacer a continuación

Verifique que un certificado esté en el almacén de claves ejecutando los siguientes mandatos que imprimen los detalles:

```
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. Definir la política de cola

Acerca de esta tarea

Después de crear el gestor de colas y preparar interceptores para interceptar mensajes y acceder a claves de cifrado, podemos comenzar a definir políticas de protección en `QM_VERIFY_AMS` mediante el mandato `setmqspl`. Consulte setmqspl para obtener más información sobre este mandato. Cada nombre de política debe ser el mismo que el nombre de cola al que se debe aplicar.

Ejemplo

Este es un ejemplo de una política definida para la cola `TEST.Q`. En este ejemplo, el usuario `alice` firma los mensajes utilizando el algoritmo Deprecated SHA1 y los cifra utilizando el algoritmo AES de 256 bits. `alice` es el único emisor válido y bob es el único receptor de los mensajes en esta cola:

```
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: Los DN coinciden exactamente con los especificados en el certificado del usuario respectivo de la base de datos de claves.

Qué hacer a continuación

Para verificar la política que ha definido, emita el mandato siguiente:

```
dspmqspl -m QM_VERIFY_AMS
```

Para imprimir los detalles de la política como un conjunto de mandatos `setmqspl`, utilice el distintivo `-export`. Esto permite almacenar políticas ya definidas:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Probar la configuración

Acerca de esta tarea

Puede ejecutar programas diferentes bajo usuarios diferentes para verificar si la aplicación se ha configurado debidamente.

Procedimiento

1. Cambie al directorio que contiene los ejemplos. Si MQ está instalado en una ubicación no predeterminada, puede estar en otro lugar.

```
cd /opt/mqm/samp/bin
```

2. Cambie el usuario de modo que se ejecute como usuario `alice`

```
su alice
```

3. Como usuario `alice`, coloque un mensaje utilizando una aplicación de ejemplo:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. Escriba el texto del mensaje y pulse Intro.
5. Deje de ejecutarse como el usuario `alice`

```
exit
```

6. Cambie el usuario de modo que se ejecute como usuario bob

```
su bob
```

7. Como usuario bob, obtenga un mensaje utilizando una aplicación de ejemplo:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Resultados

Si la aplicación se ha configurado debidamente para ambos usuarios, el mensaje del usuario `alice` se visualiza cuando bob ejecuta la aplicación de obtención.

8. Probar el cifrado

Acerca de esta tarea

Para verificar que el cifrado se realiza según lo esperado, cree una cola de alias que haga referencia a la cola original TEST.Q. Esta cola de alias no tendrá ninguna política de seguridad, por lo que ningún usuario tendrá la información para descifrar el mensaje y, por lo tanto, se mostrarán los datos cifrados.

Procedimiento

1. Utilizando el mandato **runmqsc** en el gestor de colas QM_VERIFY_AMS, cree una cola de alias.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Otorgue a bob el acceso para examinar desde la cola de alias

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. Como usuario `alice`, coloque otro mensaje utilizando una aplicación de ejemplo al igual que antes:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. Como usuario bob, examine el mensaje utilizando una aplicación de ejemplo utilizando la cola de alias esta vez:

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. Como usuario bob, obtenga el mensaje utilizando una aplicación de ejemplo desde la cola local:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Resultados

La salida de la aplicación amqsbcg mostrará los datos cifrados que hay en la cola, lo que demuestra que el mensaje se ha cifrado.

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 *Colocación en cola local de mensajes protegidos por privacidad para AMS en z/OS*

En este ejemplo se describen las políticas de Advanced Message Security y los certificados necesarios para enviar y recuperar los mensajes protegidos por privacidad a y desde una cola local para las aplicaciones de transferencia y obtención. Los mensajes protegidos por privacidad están firmados y cifrados.

El gestor de colas de ejemplo y el gestor local son los siguientes:

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

Se utilizan estos usuarios:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

Los pasos para configurar este escenario son:

Crear el certificado de usuario

En este ejemplo, se necesitan dos certificados de usuario. Estos son el certificado del usuario emisor necesario para firmar mensajes y el certificado del usuario receptor necesario para cifrar y descifrar los datos de mensajes. El usuario emisor es 'TELLER5' y el usuario receptor es 'FINADM2'.

El certificado de la CA (Certificate Authority) también es necesario.El certificado de la CA es el certificado de la autoridad que ha emitido el certificado del usuario. Puede ser una cadena de certificados. Si es así, todos los certificados de la cadena serán necesarios en el conjunto de claves del usuario de la tarea Advanced Message Security, en este caso el usuario WMQBNK6.

Se puede crear un certificado de CA mediante el mandato RACDCERT de RACF. Este certificado se utiliza para emitir los certificados de usuario. Por ejemplo:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Este mandato RACDCERT crea un certificado de CA que se puede utilizar para emitir un certificado de usuario para el usuario 'TELLER5' y el usuario 'FINADM2'. Por ejemplo:

```
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)
```

La instalación tendrá procedimientos para seleccionar o crear un certificado de CA, así como procedimientos para emitir los certificados y distribuirlos en los sistemas relevantes.

Cuando se exportan e importan estos certificados, Advanced Message Security requieren:

- El certificado de CA (cadena).
- El certificado del usuario emisor y su clave privada.
- El certificado del usuario receptor y su clave privada.

Si está utilizando RACF, se puede utilizar el mandato RACDCERT EXPORT para exportar los certificados a un conjunto de datos y se puede utilizar el mandato RACDCERT ADD para importar certificados desde el conjunto de datos. Para obtener más información sobre estos y otros mandatos RACDCERT, consulte RACDCERT (Manage RACF digital certificates) en la publicación *z/OS: Security Server RACF Command Language Reference*.

En este caso, los certificados son necesarios en el sistema z/OS que ejecuta el gestor de colas BNK6.

Cuando se importan los certificados en el sistema z/OS que ejecuta BNK6, los certificados de usuarios requieren el atributo TRUST. El mandato RACDCERT ALTER se puede utilizar para añadir el atributo TRUST al certificado. Por ejemplo:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Conectar los certificados a los conjuntos de claves relevantes

Una vez creados o importados los certificados necesarios, deben conectarse a los conjuntos de claves de usuario adecuados en el sistema z/OS que ejecuta BNK6. Para crear los conjuntos de claves utilice el mandato RACDCERT ADDRING:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Esto crea un conjunto de claves para el usuario de la tarea Advanced Message Security y un conjunto de claves para el usuario emisor y el usuario receptor. Tenga en cuenta que el nombre de conjunto de claves `drq.ams.keyring` es obligatorio y que el nombre distingue entre mayúsculas y minúsculas.

Una vez creados los conjuntos de claves, se pueden conectar los certificados relevantes.

```
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))
```

Los certificados del usuario emisor y el usuario receptor se deben conectar como DEFAULT. Si el usuario tiene más de un certificado en drq.ams.keyring, se utiliza el certificado predeterminado para fines de firma y descifrado.

El certificado del usuario emisor también debe estar conectado al conjunto de claves del usuario de la tarea Advanced Message Security con USAGE(SITE). Esto es debido a que la tarea Advanced Message Security necesita la clave pública del receptor cuando cifra los datos de mensajes. USAGE(SITE) impide que se pueda acceder a la clave privada desde el conjunto de claves.

Advanced Message Security no reconoce la creación y modificación de los certificados hasta que se ha detenido y reiniciado o hasta que se emita el mandato z/OS **MODIFY** para renovar la configuración de certificados de Advanced Message Security. Por ejemplo:

```
F BNK6AMSM,REFRESH KEYRING
```

Crear la política de Advanced Message Security

En este ejemplo, los mensajes protegidos por privacidad los coloca en la cola FIN.XFER.Q8 una aplicación que se ejecuta como el usuario 'TELLER5' y los recupera de la misma cola una aplicación que se ejecuta como el usuario 'FINADM2', por lo tanto, solo se requiere una política de Advanced Message Security.

El programa de utilidad política de seguridad de mensaje (CSQ0UTIL) Advanced Message Security se crean utilizando el programa de utilidad CSQ0UTIL que se describe en El programa de utilidad de política de seguridad de mensajes (CSQ0UTIL).

Utilice el programa de utilidad CSQ0UTIL para ejecutar los mandatos siguientes:

```
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
```

En esta política, el gestor de colas se identifica como BNK6. El nombre de política y la cola asociada es FIN.XFER.Q8. El algoritmo que se utiliza para generar la firma del remitente es Deprecated SHA1, y el nombre distinguido (DN) del usuario remitente es 'CN=Teller5,O=BCO,C=US', y el usuario destinatario es 'CN=FinAdm2,O=BCO,C=US'. El algoritmo que se utiliza para cifrar los datos del mensaje es Deprecated 3DES.

Después de definir la política, reinicie el gestor de colas BNK6, o utilice el mandato z/OS **MODIFY** para renovar la configuración de políticas de Advanced Message Security. Por ejemplo:

```
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 *Colocación en cola remota de mensajes protegidos por privacidad para AMS en z/OS*

En este ejemplo se describen las políticas de Advanced Message Security y los certificados necesarios para enviar y recuperar los mensajes protegidos por privacidad a y desde colas gestionadas por dos gestores de colas diferentes. Los dos gestores de colas pueden estar ejecutándose en el mismo sistema

z/OS, en sistemas z/OS diferentes o un gestor de colas puede estar en un sistema distribuido que ejecuta Advanced Message Security.

Los gestores de colas de ejemplo y las colas son:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Nota: En este ejemplo, BNK6 y BNK7 son los gestores de colas que se ejecutan en diferentes sistemas z/OS con el mismo nombre.

Se utilizan estos usuarios:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

Los pasos para configurar este escenario son:

Crear el certificado de usuario

En este ejemplo, se necesitan dos certificados de usuario. Estos son el certificado del usuario emisor necesario para firmar mensajes y el certificado del usuario receptor necesario para cifrar y descifrar los datos de mensajes. El usuario emisor es 'TELLER5' y el usuario receptor es 'FINADM2'.

El certificado de la CA (Certificate Authority) también es necesario.El certificado de la CA es el certificado de la autoridad que ha emitido el certificado del usuario. Puede ser una cadena de certificados. Si es así, todos los certificados de la cadena serán necesarios en el conjunto de claves del usuario de la tarea Advanced Message Security, en este caso el usuario WMQBNK7.

Se puede crear un certificado de CA mediante el mandato RACDCERT de RACF. Este certificado se utiliza para emitir los certificados de usuario. Por ejemplo:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Este mandato RACDCERT crea un certificado de CA que se puede utilizar para emitir un certificado de usuario para el usuario 'TELLER5' y el usuario 'FINADM2'. Por ejemplo:

```
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)
```

La instalación tendrá procedimientos para seleccionar o crear un certificado de CA, así como procedimientos para emitir los certificados y distribuirlos en los sistemas relevantes.

Cuando se exportan e importan estos certificados, Advanced Message Security requieren:

- El certificado de CA (cadena).
- El certificado del usuario emisor y su clave privada.
- El certificado del usuario receptor y su clave privada.

Si está utilizando RACF, se puede utilizar el mandato RACDCERT EXPORT para exportar los certificados a un conjunto de datos y se puede utilizar el mandato RACDCERT ADD para importar certificados desde el conjunto de datos.

Para obtener más información sobre estos y otros mandatos RACDCERT, consulte RACDCERT (Manage RACF digital certificates) en la publicación *z/OS: Security Server RACF Command Language Reference*.

En este caso, los certificados son necesarios en el sistema z/OS que ejecuta el gestor de colas BNK6 y BNK7.

En este ejemplo, los certificados de emisor y receptor se deben importar al sistema z/OS que ejecuta BNK6 y los certificados de CA se deben importar al sistema z/OS que ejecuta BNK7. Cuando se han importado los certificados, los certificados de usuario requieren el atributo TRUST. El mandato RACDCERT ALTER se puede utilizar para añadir el atributo TRUST al certificado. Por ejemplo:

En BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

En BNK7:

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Conectar los certificados a los conjuntos de claves relevantes

Una vez creados o importados los certificados necesarios, deben conectarse a los conjuntos de claves de usuario adecuados en los sistemas z/OS que ejecutan BNK6 y BNK7.

Para crear los conjuntos de claves utilice el mandato RACDCERT ADDRING:

En BNK6:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Esto crea un conjunto de claves para el usuario de la tarea de Advanced Message Security y un conjunto de claves para el usuario emisor en BNK6. Tenga en cuenta que el nombre del conjunto de claves drq.ams.keyring es obligatorio y que el nombre distingue entre mayúsculas y minúsculas.

En BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Esto crea un conjunto de claves para el usuario de la tarea de Advanced Message Security y un conjunto de claves para el usuario receptor en BNK7.

Una vez creados los conjuntos de claves, se pueden conectar los certificados relevantes.

En 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))
```

En 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))
```

Los certificados del usuario emisor y el usuario receptor se deben conectar como DEFAULT. Si el usuario tiene más de un certificado en drq.ams.keyring, se utiliza el certificado predeterminado para fines de cifrado y descifrado.

En BNK6, el certificado del usuario emisor también debe estar conectado al conjunto de claves del usuario de la tarea Advanced Message Security con USAGE(SITE). Esto es debido a que la tarea Advanced Message Security necesita la clave pública del receptor cuando cifra los datos de mensajes. USAGE(SITE) impide que se pueda acceder a la clave privada desde el conjunto de claves.

Advanced Message Security no reconoce la creación y modificación de los certificados hasta que se ha detenido y reiniciado o hasta que se emita el mandato z/OS **MODIFY** para renovar la configuración de certificados de Advanced Message Security. Por ejemplo:

En BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

En BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Crear las políticas de Advanced Message Security

En este ejemplo, los mensajes protegidos por privacidad los coloca en la cola FIN.XFER.Q7 en BNK6 una aplicación que se ejecuta como el usuario 'TELLER5' y los recupera de la cola local FIN.RCPT.Q7 en BNK7 una aplicación que se ejecuta como el usuario 'FINADM2' , por lo tanto, se requieren dos políticas de Advanced Message Security.

El programa de utilidad política de seguridad de mensaje (CSQ0UTIL) Advanced Message Security se crean utilizando el programa de utilidad CSQ0UTIL que se describe en El programa de utilidad de política de seguridad de mensajes (CSQ0UTIL).

Utilice el programa de utilidad CSQ0UTIL para ejecutar el mandato siguiente para definir una política de privacidad para la cola remota en 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
```

En esta política, el gestor de colas se identifica como BNK6. El nombre de política y la cola asociada es FIN.XFER.Q7. El algoritmo que se utiliza para generar la firma del remitente es Deprecated SHA1, el nombre distinguido (DN) del usuario emisor es 'CN=Teller5,O=BCO,C=US' y el usuario destinatario es 'CN=FinAdm2,O=BCO,C=US'. El algoritmo que se utiliza para cifrar los datos del mensaje es Deprecated 3DES.

Utilice también el programa de utilidad CSQ0UTIL para ejecutar el mandato siguiente para definir una política de privacidad para la cola local en 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
```

En esta política, el gestor de colas se identifica como BNK7. El nombre de política y la cola asociada es FIN.RCPT.Q7. El algoritmo esperado para la firma del remitente es Deprecated SHA1, se espera que el nombre distinguido (DN) del usuario emisor sea 'CN=Teller5,O=BCO,C=US' y que el usuario destinatario sea 'CN=FinAdm2,O=BCO,C=US'. El algoritmo que se utiliza para descifrar los datos del mensaje es Deprecated 3DES.

Después de definir dos políticas, reinicie los gestores de colas BNK6 y BNK7, o utilice el mandato z/OS **MODIFY** para renovar la configuración de política de Advanced Message Security. Por ejemplo:

En BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

En BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

Guía de inicio rápido para AMS con clientes Java

Utilice esta guía para aprender a configurar rápidamente Advanced Message Security para proporcionar seguridad de mensajes para las aplicaciones Java que se conectan utilizando enlaces de cliente. Cuando lo haya completado, habrá creado un almacén de claves para verificar las identidades de usuario y habrá definido políticas de firma/cifrado para su gestor de colas.

Antes de empezar

Asegúrese de que tiene los componentes adecuados instalados tal como se describe en "Guía de inicio rápido para AMS en plataformas Windows" en la página 619 o "Guía de inicio rápido para AMS en AIX and Linux" en la página 625.

1. Crear un gestor de colas y una cola

Acerca de esta tarea

Todos los ejemplos siguientes utilizan una cola denominada TEST.Q para pasar mensajes entre aplicaciones. Advanced Message Security utiliza interceptores para firmar y cifrar mensajes en el momento en que los mensajes entran en la infraestructura de IBM MQ a través de la interfaz estándar de IBM MQ. La configuración básica se realiza en IBM MQ y se define en los pasos siguientes.

Procedimiento

1. Crear un gestor de colas

   ```
   crtmqm QM_VERIFY_AMS
   ```

2. Inicie el gestor de colas

   ```
   strmqm QM_VERIFY_AMS
   ```

3. Cree e inicie un escucha especificando los mandatos siguientes en **runmqsc** para el gestor de colas QM_VERIFY_AMS

   ```
   DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
   ```

   ```
   START LISTENER(AMS.LSTR)
   ```

4. Cree un canal para que se conecten las aplicaciones especificando el siguiente mandato en **runmqsc** para el gestor de colas QM_VERIFY_AMS

   ```
   DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
   ```

5. Cree una cola denominada TEST.Q especificando el siguiente mandato en **runmqsc** para el gestor de colas QM_VERIFY_AMS

   ```
   DEFINE QLOCAL(TEST.Q)
   ```

Resultados

Si el procedimiento se ha ejecutado correctamente, el siguiente mandato especificado en **runmqsc** muestra detalles sobre TEST.Q:

```
DISPLAY Q(TEST.Q)
```

2. Crear y autorizar usuarios

Acerca de esta tarea

En este caso de ejemplo existen dos usuarios: `alice`, el emisor, y bob, el receptor. Para utilizar la cola de aplicación, estos usuarios deben tener autorización para utilizarla. Asimismo, para utilizar correctamente las políticas de protección definidas en este caso de ejemplo, estos usuarios deben tener acceso a algunas colas del sistema. Para obtener más información sobre el mandato **setmqaut** , consulte **setmqaut**.

Procedimiento

1. Cree los dos usuarios, tal como se describe en la **Guía de inicio rápido** (Windows o AIX and Linux) correspondiente a su plataforma.
2. Autorice a los usuarios para conectarse con el gestor de colas y para trabajar con la cola

   ```
   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. También debe permitir que los dos usuarios examinen la cola de políticas del sistema y coloquen mensajes en la cola de errores.

   ```
   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
   ```

   Atención: IBM MQ optimiza el rendimiento almacenando en memoria caché las políticas para que no tenga que examinar los registros para obtener detalles de política en el SYSTEM.PROTECTION.POLICY.QUEUE en todos los casos.

   IBM MQ no almacena en memoria caché todas las políticas disponibles. Si hay un número alto de políticas, IBM MQ almacena en memoria caché un número limitado de políticas. Por lo tanto, si el gestor de colas tiene un número bajo de políticas definidas, no es necesario proporcionar la opción de examinar a SYSTEM.PROTECTION.POLICY.QUEUE.

   Sin embargo, debe otorgar autorización para examinar esta cola, en caso de que haya un gran número de políticas definidas, o si está utilizando clientes antiguos. El sistema SYSTEM.PROTECTION.ERROR.QUEUE se utiliza para colocar mensajes de error generados por el código AMS. La autorización de colocación sobre esta cola sólo se comprueba cuando se intenta colocar un mensaje de error en la cola. La autorización de colocación sobre la cola no se comprueba cuando intenta transferir u obtener un mensaje de una cola protegida por AMS.

Resultados

Se crean los usuarios y se les otorgan las autorizaciones necesarias.

Qué hacer a continuación

Para verificar si los pasos se han realizado correctamente, utilice los ejemplos `JmsProducer` y `JmsConsumer`, tal como se describe en la sección "7. Probar la configuración" en la página 647.

3. Crear la base de datos de claves y certificados

Acerca de esta tarea

Para cifrar el mensaje para el los interceptores es necesario la clave pública de los usuarios emisores. Por lo tanto, se deben crear la base de datos de claves de identidades de usuario que están correlacionadas con claves públicas y privadas. En el sistema real, donde los usuarios y las aplicaciones están distribuidos en varios sistemas, cada usuario tendría su propio almacén de claves privado. De forma similar, en esta guía, creamos bases de datos de claves para `alice` y bob y compartimos los certificados de usuario entre ellos.

Nota: En esta guía, utilizamos aplicaciones de ejemplo escritas en Java que se conectan mediante enlaces de cliente. Si tiene previsto utilizar aplicaciones Java utilizando enlaces locales o las aplicaciones C, debe crear un almacén de claves CMS y los certificados mediante el mandato **runmqakm**. Para obtener más información, consulte "Guía de inicio rápido para AMS en plataformas Windows" en la página 619 y "Guía de inicio rápido para AMS en AIX and Linux" en la página 625.

Procedimiento

1. Cree un directorio en el que crear el almacén de claves, por ejemplo `/home/alice/.mqs`. Es posible que desee crearlo en el mismo directorio que utiliza la Guía de inicio rápido para su plataforma. Para obtener más información, consulte "Guía de inicio rápido para AMS en plataformas Windows" en la página 619 y "Guía de inicio rápido para AMS en AIX and Linux" en la página 625.

   Nota: Este directorio se conoce como *keystore-dir* en los siguientes pasos
2. Cree un nuevo almacén de claves y un certificado que identifique al usuario `alice` para utilizarlo en el cifrado

   Nota: El mandato **keytool** es parte del 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:

   - Si *keystore-dir* contiene espacios, debe escribir el nombre completo del almacén de claves entre comillas
   - Se recomienda utilizar una contraseña fuerte para proteger el almacén de claves.
   - A los efectos de esta guía, utilizamos un certificado autofirmado que se puede crear sin utilizar una entidad emisora de certificados. Para los sistemas de producción, se recomienda no utilizar certificados autofirmados, sino certificados firmados por una entidad emisora de certificados.
   - El parámetro **alias** especifica el nombre para el certificado, que los interceptores buscarán para recibir la información necesaria.
   - El parámetro **dname** especifica los detalles del **Nombre distinguido** (DN), que debe ser exclusivo para cada usuario.
3. En AIX and Linux, asegúrese de que el almacén de claves sea legible

   ```
   chmod +r keystore-dir/keystore.jks
   ```
4. Repita los pasos del 1 al 4 para el usuario bob

Resultados

Los dos usuarios `alice` y bob tienen cada uno un certificado autofirmado.

4. Crear keystore.conf

Acerca de esta tarea

Debe apuntar los interceptores de Advanced Message Security al directorio donde residen las bases de datos y certificados. Esto se realiza mediante el archivo `keystore.conf`, que contiene esa información como texto sin formato. Cada usuario debe tener un archivo `keystore.conf` separado. Este paso debe realizarse para `alice` y bob.

Ejemplo

Para este caso de ejemplo, el contenido de `keystore.conf` para `alice` es como el siguiente:

```
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 caso de ejemplo, el contenido de `keystore.conf` para bob es como el siguiente:

```
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:

- La vía de acceso del archivo de almacén de claves se debe especificar sin ninguna extensión de archivo.
- Si ya tiene un archivo `keystore.conf` porque ha seguido las instrucciones de la Guía de inicio rápido (Windows o AIX and Linux), puede editar el archivo existente para añadir estas líneas.
- Para obtener más información, consulte “Estructura del archivo de configuración del almacén de claves (keystore.conf) para AMS” en la página 656.

5. Compartir certificados

Acerca de esta tarea

Comparta los certificados entre los dos almacenes de claves para que cada usuario pueda identificar correctamente al otro. Esto se realiza extrayendo el certificado de cada usuario e importándolo en el almacén de claves de otro usuario.

Importante: Los términos *extract* y *export* se utilizan de forma diferente en los distintos mandatos de gestión de certificados.

- El mandato IBM Global Security Kit (GSKit) **runmqakm** utiliza el término *extract* para hacer referencia al proceso de copiar sólo la parte pública de un certificado de un almacén de claves, y el término *export* para hacer referencia al proceso de copiar certificados y sus claves públicas y privadas asociadas de un almacén de claves a otro.
- El mandato Java **keytool** , V 9.4.0 V 9.4.0 y el mandato IBM MQ **runmqktool**, utilizan el término *export* para hacer referencia al proceso de copiar sólo la parte pública de un certificado de un almacén de claves.

Esta distinción es importante ya que el uso de *export* de forma incorrecta puede comprometer la aplicación al exponer su clave privada. Debido a que la distinción es tan importante, la documentación de IBM MQ utiliza estos términos de forma coherente. Por estas razones, el procedimiento siguiente hace referencia a la *extracción* de certificados utilizando la opción `exportcert` en el mandato **keytool** .

Procedimiento

1. Extraiga el certificado que identifica a `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 el certificado que identifica a `alice` al almacén de claves que utilizará bob. Cuando se le solicite, indique que no confía en este 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 los pasos para bob

Resultados

Los dos usuarios `alice` y bob pueden identificar ahora correctamente al otro mediante la creación y compartición de certificados autofirmados.

Qué hacer a continuación

Verifique que un certificado esté en el almacén de claves ejecutando los siguientes mandatos que imprimen los detalles:

```
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. Definir la política de cola

Acerca de esta tarea

Después de crear el gestor de colas y preparar interceptores para interceptar mensajes y acceder a claves de cifrado, podemos comenzar a definir políticas de protección en `QM_VERIFY_AMS` mediante el mandato `setmqspl`. Consulte setmqspl para obtener más información sobre este mandato. Cada nombre de política debe ser el mismo que el nombre de cola al que se debe aplicar.

Ejemplo

Esto es un ejemplo de una política definida para la cola `TEST.Q`, firmada por el usuario `alice` mediante el algoritmo Deprecated SHA1 y cifrada utilizando el algoritmo AES de 256 bits para el usuario 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: Los DN coinciden exactamente con los especificados en el certificado del usuario respectivo de la base de datos de claves.

Qué hacer a continuación

Para verificar la política que ha definido, emita el mandato siguiente:

```
dspmqspl -m QM_VERIFY_AMS
```

Para mostrar los detalles de la política como un conjunto de mandatos `setmqspl`, utilice el distintivo `-export`. Esto permite almacenar políticas ya definidas:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Probar la configuración

Antes de empezar

Asegúrese de que la versión de Java que está utilizando tiene instalados los archivos de política JCE sin restricciones.

Nota: La versión de Java proporcionada en la instalación de IBM MQ ya tiene estos archivos de política. Puede encontrarse en *MQ_INSTALLATION_PATH*`/java/bin`.

Acerca de esta tarea

Puede ejecutar programas diferentes bajo usuarios diferentes para verificar si la aplicación se ha configurado debidamente. Para obtener más información sobre la ejecución de programas bajo distintos usuarios, consulte "Guía de inicio rápido para AMS en plataformas Windows" en la página 619 y "Guía de inicio rápido para AMS en AIX and Linux" en la página 625.

Procedimiento

1. Para ejecutar estas aplicaciones de ejemplo JMS, utilice el valor de CLASSPATH correspondiente a su plataforma, tal como se muestra en Variables de entorno utilizadas por las IBM MQ classes for JMS para asegurarse de que se incluye el directorio de ejemplos.
2. Como usuario `alice`, coloque un mensaje utilizando una aplicación de ejemplo, conectarse como un cliente:

   ```
   java JmsProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

3. Como usuario bob, obtenga un mensaje utilizando una aplicación de ejemplo, conectarse como un cliente:

   ```
   java JmsConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

Resultados

Si la aplicación se ha configurado debidamente para ambos usuarios, el mensaje del usuario `alice` se visualiza cuando bob ejecuta la aplicación de obtención.

Protección de colas remotas en AMS

Para proteger completamente las colas remotas, deben establecerse políticas en la cola remota y en la cola local a las que se transmiten los mensajes.

Cuando se pone un mensaje en una cola remota, Advanced Message Security intercepta la operación y procesa el mensaje según un conjunto de políticas definido para la cola remota. Por ejemplo, para una política de cifrado, el mensaje se cifra antes de que se pase a IBM MQ para su proceso. Después de procesar el mensaje, Advanced Message Security lo coloca en una cola remota, IBM MQ coloca el mensaje en la cola de transmisión asociada y lo reenvía al gestor de colas de destino y cola de destino.

Cuando la operación GET se realiza en la cola local, Advanced Message Security intenta descifrar el mensaje de acuerdo con la política definida en la cola local. Para que la operación sea satisfactoria, la política utilizada para descifrar el mensaje debe ser la misma que la utilizada para cifrarlo. Cualquier discrepancia provocará que se rechace el mensaje.

Si por cualquier motivo las políticas no se puede definir al mismo tiempo, se proporciona un mecanismo de despliegue gradual. La política se puede definir en una cola local con el distintivo de tolerancia

activado, el cual indica que se pase por alto la política asociada a una cola cuando el intento de recuperar un mensaje de la cola afecte a un mensaje que no tenga definida la política de seguridad. En este caso, GET intentará descifrar el mensaje, pero permitirá la entrega de los mensaje no cifrados. De esta forma se pueden definir políticas en colas remotas después de proteger (y probar) las colas locales.

Recuerde: Elimine el distintivo de tolerancia una vez que concluya el despliegue de Advanced Message Security.

Referencia relacionada

setmqspl (establecer política de seguridad)

Direccionamiento de mensajes protegidos con AMS utilizando IBM Integration Bus

Advanced Message Security puede proteger los mensajes en una infraestructura donde está instalado IBM Integration Buso WebSphere Message Broker 8.0.0.1 (o posterior). Debe comprender la naturaleza de ambos productos antes de aplicar la seguridad en el entorno de IBM Integration Bus.

Acerca de esta tarea

Advanced Message Security proporciona seguridad global para la carga útil del mensaje. Esto significa que sólo los interlocutores especificados como remitentes y destinatarios válidos de un mensaje pueden emitirlo o recibirlo. Esto significa que para proteger los mensajes que fluyen por IBM Integration Bus, puede permitir que IBM Integration Bus procese los mensajes sin conocer su contenido (Caso de ejemplo 1) o convertirlo en un usuario con autorización para recibir y enviar mensajes (Caso de ejemplo 2).

Caso de ejemplo 1 - El bus de integración no puede ver el contenido del mensaje

Antes de empezar

Debe tener su IBM Integration Bus conectado a un gestor de colas existente. Sustituya *QMgrName* por este nombre de gestor de colas existente en los mandatos siguiente.

Acerca de esta tarea

En este caso de ejemplo, Alice coloca un mensaje protegido en una cola de entrada `QIN`. Basándose en la propiedad de mensaje `routeTo`, el mensaje se direcciona a *bob's* (`QBOB`),[1](`QCECIL`), o la cola predeterminada (`QDEF`). El direccionamiento es posible porque Advanced Message Security sólo protege la carga útil del mensaje y no sus cabeceras y propiedades, que permanecen desprotegidos y pueden ser leídos por IBM Integration Bus. Advanced Message Security es utilizado sólo por *alice*, *bob* y *cecil*. No es necesario instalarlo ni configurarlo para IBM Integration Bus.

IBM Integration Bus recibe el mensaje protegido desde la cola del alias no protegida para evitar cualquier intento de descifrar el mensaje. Si desea utilizar la cola protegida directamente, el mensaje debe colocarse en la cola de mensajes no entregados como imposible de descifrar. IBM Integration Bus direcciona el mensaje, que llega a la cola de destino sin modificar. Por lo tanto, es todavía firmado por el autor original (tanto *bob* como *cecil* sólo aceptarán mensajes enviados por *alice*) y está protegido como antes (sólo *bob* y *cecil* pueden leerlo). IBM Integration Bus coloca el mensaje direccionado en un alias no protegido. Los destinatarios recuperan el mensaje de una cola de salida protegida donde AMS descifrará de forma transparente el mensaje.

Procedimiento

1. Configure *alice*, *bob* y *cecil* para que utilicen Advanced Message Security, tal como se describe en la **Guía de inicio rápido** (Windows o AIX).

   Asegúrese de que los pasos siguientes se hayan completado:

   - Crear y autorizar usuarios
   - Crear la base de datos de claves y certificados
   - Crear keystore.conf

[1] cecil's

2. Proporcione el certificado de *alice* a *bob* y *cecil*, de modo que *alice* pueda ser identificada por ellos durante la comprobación de firmas digitales en los mensajes.

   Hágalo extrayendo el certificado que identifica a *alice* a un archivo externo y, después, añadiendo el certificado extraído a los almacenes de claves *de bob* y *de cecil*. Es importante que utilice el método descrito en la tarea 5 de **. Compartir certificados** en la **Guía de inicio rápido** (Windows o AIX).
3. Proporcione los certificados de *bob* y *cecil* a *alice*, con lo que *alice* podrá enviar mensajes cifrados a *bob* y *cecil*.

   Hágalo utilizando el método especificado en el paso anterior.
4. En el gestor de colas, defina las colas locales denominadas QIN, QBOB, QCECIL y QDEF.

   ```
   DEFINE QLOCAL(QIN)
   ```
5. Configure la política de seguridad de la cola QIN para una configuración elegible. Utilice la misma configuración para las colas QBOB, QCECIL y 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 caso de ejemplo presupone la política de seguridad en la que *alice* es el único emisor autorizado y *bob* y *cecil* son los destinatarios.
6. Defina las colas de alias AIN, ABOB y ACECIL que hacen referencia a las colas locales QIN, QBOB y QCECIL, respectivamente.

   ```
   DEFINE QALIAS(AIN) TARGET(QIN)
   ```
7. Verifique que la configuración de seguridad para los alias especificados en el paso anterior no está presente; de lo contrario, establezca su política en NONE.

   ```
   dspmqspl -m QMgrName -p AIN
   ```
8. En IBM Integration Bus, cree un flujo de mensajes para direccionar los mensajes que llegan a la cola de alias AIN al nodo BOB, CECIL o DEF, dependiendo de la propiedad `routeTo` del mensaje. Para ello:
   a) Cree un nodo `MQInput` denominado IN y asigne el alias AIN como su nombre de cola.
   b) Cree nodos `MQOutput` denominados BOB, CECIL y DEF y asigne las colas de alias ABOB, ACECIL y ADEF como sus nombres de colas respectivos.
   c) Cree un nodo de ruta y asígnele el nombre TEST.
   d) Conecte el nodo IN al terminal de entrada del nodo TEST.
   e) Cree los terminales de salida bob y `cecil` para el nodo TEST.
   f) Conecte el terminal de salida bob al nodo BOB.
   g) Conecte el terminal de salida `cecil` al nodo CECIL.
   h) Conecte el nodo DEF al terminal de salida predeterminado.
   i) Aplique las reglas siguientes:

      ```
      $Root/MQRFH2/usr/routeTo/text()="bob"
      ```

      ```
      $Root/MQRFH2/usr/routeTo/text()="cecil"
      ```
9. Despliegue el flujo de mensajes en el componente de ejecución de IBM Integration Bus.
10. Ejecutándose como el usuario `Alice`, coloque un mensaje que también contenga una propiedad de mensaje denominada `routeTo` con un valor bob o `cecil`. Para ello, ejecute la aplicación de ejemplo **amqsstm**.

```
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. Ejecutándose como el usuario *bob*, recupere el mensaje de la cola QBOB utilizando la aplicación de ejemplo **amqsget**.

Resultados

Cuando *alice* coloca un mensaje en la cola QIN, el mensaje queda protegido. IBM Integration Bus recupera el mensaje con el formato protegido del alias AIN. IBM Integration Bus decide adónde direccionar el mensaje examinando la propiedad routeTo, la cual no está cifrada, como ocurre con todas las propiedades. IBM Integration Bus coloca el mensaje en el alias no protegido apropiado para evitar su protección ulterior. Cuando *bob* o *cecil* reciben el mensaje de la cola, se descifra el mensaje y se verifica la firma digital.

Caso de ejemplo 2 - El bus de integración puede ver el contenido del mensaje

Acerca de esta tarea

En este ejemplo, un grupo de usuarios está autorizado para enviar mensajes a IBM Integration Bus. Otro grupo está autorizado para recibir los mensajes que ha creado IBM Integration Bus. La transmisión entre las partes y IBM Integration Bus no puede ser interceptada.

Tenga en cuenta que IBM Integration Bus lee las políticas de protección y los certificados una sola vez, por lo que debe volver a cargar el grupo de ejecución después de realizar cualquier actualización en las políticas de protección para que los cambios surtan efecto.

```
mqsireload execution-group-name
```

Si se considera que IBM Integration Bus es un interlocutor autorizado con permiso para leer o firmar la carga útil del mensaje, debe configurar Advanced Message Security para el usuario encargado de iniciar el servicio de IBM Integration Bus. Tenga en cuenta que no es necesariamente el mismo usuario que realiza operaciones PUT o GET para mensajes de las colas ni el usuario que crea y despliega aplicaciones de IBM Integration Bus.

Procedimiento

1. Configure *alice*, *bob*, *cecil* y *dave* y el usuario de servicio de IBM Integration Bus para que se utilice Advanced Message Security como se describe en la **Guía de inicio rápido** (Windows o AIX).

   Asegúrese de que los pasos siguientes se hayan completado:

   - Crear y autorizar usuarios
   - Crear la base de datos de claves y certificados
   - Crear keystore.conf
2. Proporcione los certificados de *alice*, *bob*, *cecil* y *dave* al usuario de servicio IBM Integration Bus.

   Hágalo extrayendo cada uno de los certificados que identifica a *alice*, *bob*, *cecil* y *dave* en archivos externos y, después, añadiendo los certificados extraídos al almacén de claves de IBM Integration Bus. Es importante que utilice el método descrito en la tarea 5 de **. Compartir certificados** en la **Guía de inicio rápido** (Windows o AIX).
3. Proporcione el certificado del usuario de servicio de IBM Integration Bus a *alice*, *bob*, *cecil* y *dave*.

Hágalo utilizando el método especificado en el paso anterior.

Nota: *Alice* y *Bob* necesitan el certificado del usuario de servicio de IBM Integration Bus para cifrar los mensajes correctamente. El usuario de servicio de IBM Integration Bus necesita los certificados de *Alice* y *Bob* para verificar los autores de los mensajes. El usuario de servicio de IBM Integration Bus necesita los certificados de *Cecil* y *Dave* para cifrar los mensajes destinados a ellos. *cecil* y *dave* necesitan el certificado del usuario de servicio de IBM Integration Bus para verificar si el mensaje procede de IBM Integration Bus.

4. Defina una cola local denominada `IN` y defina la política de seguridad con *alice* y *bob* especificados como autores y el usuario de servicio para IBM Integration Bus especificado como destinatario:

```
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 una cola local llamada `OUT` y defina la política de seguridad con el usuario de servicio para IBM Integration Bus especificado como autor, y *cecil* y *dave* especificados como destinatarios:

```
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. En IBM Integration Bus, cree un flujo de mensajes con un nodo `MQInput` y `MQOutput`. Configure el nodo `MQInput` para utilizar la cola `IN` y el nodo `MQOutput` para utilizar la cola `OUT`.
7. Despliegue el flujo de mensajes en el componente de ejecución de IBM Integration Bus.
8. Ejecutándose como el usuario *Alice* o *Bob*, coloque un mensaje en la cola `IN` utilizando la aplicación de ejemplo **amqsput**.
9. Ejecutándose como el usuario *Cecil* o *Dave*, recupere el mensaje de la cola `OUT` utilizando la aplicación de ejemplo **amqsget**.

Resultados

Los mensajes enviados por *Alice* o *Bob* a la cola de entrada `IN` están cifrados, por lo que sólo puede leerlos IBM Integration Bus. IBM Integration Bus solo acepta mensajes de *alice* y *bob* y rechaza los otros. Los mensajes aceptados se procesarán debidamente y, a continuación, se firmarán y cifrarán con las claves de *Cecil* y *Dave* antes de colocarse en la cola de salida `OUT`. Sólo lo pueden leer *Cecil* y *Dave* y se rechazarán los mensajes no firmados por IBM Integration Bus.

Utilización de Advanced Message Security con Managed File Transfer

Este caso de ejemplo explica cómo configurar Advanced Message Security para proporcionar privacidad de mensajes para los datos que se envían a través de Managed File Transfer.

Antes de empezar

Compruebe que tiene el componente Advanced Message Security instalado en la instalación de IBM MQ que aloje las colas utilizadas por Managed File Transfer que quiera proteger.

Si los agentes de Managed File Transfer se conectan en modalidad de enlaces, asegúrese de que también tiene instalado el componente IBM Global Security Kit (GSKit) en su instalación local.

Acerca de esta tarea

Cuando se interrumpe la transferencia de datos entre dos agentes de Managed File Transfer, es probable que queden datos confidenciales desprotegidos en las colas de IBM MQ subyacentes que se utilizan para gestionar la transferencia. En este caso de ejemplo, aprenderemos a configurar y utilizar Advanced Message Security para proteger estos datos en las colas de Managed File Transfer.

En este escenario, consideramos una topología simple que consta de una máquina con dos colas Managed File Transfer y dos agentes, `AGENT1` y `AGENT2`, que comparten un único gestor de colas, tal como se describe en el escenario Managed File Transfer escenario. Ambos agentes se conectan de la misma forma, ya sea en la modalidad de enlaces o en la modalidad de cliente.

1. Creación de certificados

Antes de empezar

Este escenario utiliza un modelo simple donde un usuario `ftagent` de un grupo FTAGENTS se utiliza para ejecutar los procesos Managed File Transfer Agent. Si utiliza su propios nombres de usuario y grupo, cambie los mandatos según corresponda.

Acerca de esta tarea

Advanced Message Security utiliza criptografía de clave pública para firmar o cifrar mensajes en colas protegidas.

Nota:

- Si los agentes de Managed File Transfer se ejecutan en modalidad de enlaces, los mandatos que utiliza para crear un almacén de claves de CMS (Cryptographic Message Syntax) se detallan en la **Guía de inicio rápido** (Windows o AIX) correspondiente a su plataforma.
- Si los agentes de Managed File Transfer se ejecutan en modalidad de cliente, los mandatos que necesitará para crear un JKS (almacén de claves Java) se detallan en “Guía de inicio rápido para AMS con clientes Java” en la página 642.

Procedimiento

1. Cree un certificado autofirmado para identificar al usuario `ftagent` como se describe en la Guía de inicio rápido correspondiente.
   Utilice un nombre distinguido (DN) de la siguiente manera:

   ```
   CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>
   ```

2. Cree un archivo `keystore.conf` para identificar la ubicación del almacén de claves y el certificado que contiene como se describe en la Guía de inicio rápido correspondiente.

2. Configuración de protección de mensajes

Acerca de esta tarea

Debe definir una política de seguridad para la cola de datos que utiliza AGENT2, mediante el mandato **setmqspl**. En este caso de ejemplo, se utiliza el mismo usuario para iniciar ambos agentes y, por lo tanto, el DN firmante y receptor son iguales y coinciden con el certificado que hemos generado.

Procedimiento

1. Concluya los agentes de Managed File Transfer para preparar la protección mediante el mandato **fteStopAgent**.
2. Cree una política de seguridad para proteger la cola 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. Asegúrese de que el usuario que ejecuta el proceso de Managed File Transfer Agent tiene acceso para examinar la cola de políticas del sistema y colocar mensajes en la cola de errores.

   ```
   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 los agentes de Managed File Transfer mediante el mandato **fteStartAgent**.

5. Confirme que los agentes se hayan reiniciado satisfactoriamente mediante el mandato **fteListAgents** y verifique que los agentes tengan el estado READY.

Resultados

Ahora puede enviar transferencias desde AGENT1 a AGENT2, y el contenido del archivo se transmitirá de forma segura entre los dos agentes.

Descripción general de la instalación de Advanced Message Security

Instalar el componnete Advanced Message Security en varias plataformas.

Procedimiento

- Multi
  Instalar Advanced Message Security en Multiplatforms.
- z/OS
  Instale IBM MQ Advanced for z/OS.
- z/OS
  Instale IBM MQ Advanced for z/OS Value Unit Edition.

Tareas relacionadas

Desinstalación de 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 654:

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.

Utilización de almacenes de claves y certificados con AMS

Para proporcionar protección de cifrado transparente para las aplicaciones de IBM MQ, Advanced Message Security utiliza el archivo de almacén de claves, donde se almacenan certificados de clave pública y una clave privada. En z/OS, se utiliza un conjunto de claves SAF en lugar de un archivo de almacén de claves.

En Advanced Message Security, los usuarios y las aplicaciones se representan mediante identidades de la infraestructura de claves públicas (PKI). Este tipo de identidad se utiliza para firmar y cifrar mensajes. La identidad PKI está representada por el campo **Nombre distinguido (DN)** del asunto en un certificado asociado con mensajes firmados o cifrados. Un usuario o una aplicación que desee cifrar sus mensajes debe tener acceso al archivo de almacén de claves donde se almacenan los certificados y las claves privadas y públicas asociadas.

ALW En AIX, Linux, and Windows, la ubicación del almacén de claves se proporciona en el archivo de configuración del almacén de claves, que es `keystore.conf` de forma predeterminada. Cada usuario de Advanced Message Security debe tener el archivo de configuración del almacén de claves que apunte a un archivo de almacén de claves. Advanced Message Security acepta los siguientes formatos de archivos de almacén de claves: `.kdb`, `.jceks`, `.jks`.

La ubicación predeterminada del archivo `keystore.conf` es:

- Linux IBM i AIX En IBM i, AIX and Linux: `$HOME/.mqs/keystore.conf`
- Windows En Windows: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

Si está utilizando un nombre de archivo y una ubicación de almacén de claves especificados, debe especificarlo con la variable de entorno **MQS_KEYSTORE_CONF** , tal como se muestra en los siguientes mandatos de ejemplo:

- Para Java: `java -D`*`MQS_KEYSTORE_CONF=path/filename app_name`*
- Para un cliente y servidor C:
  - Linux AIX En AIX and Linux: `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - Windows En Windows: `set MQS_KEYSTORE_CONF=`*`path\filename`*

    Nota: La vía de acceso en Windows puede y debe especificar la letra de unidad si hay más de una letra de unidad disponible.

Protección de información confidencial en el archivo `keystore.conf`

Para acceder a la información confidencial del archivo de almacén de claves, como las contraseñas, debe proporcionar señales para que IBM MQ Advanced Message Security (AMS) pueda acceder al almacén de claves y firmar y cifrar mensajes.

Debe proteger la información confidencial contenida en el archivo de configuración del almacén de claves utilizando el mandato **runamscred** proporcionado con AMS. Consulte “Configuración de la protección por contraseña de AMS para los archivos de configuración” en la página 674 para obtener detalles sobre cómo proteger los archivos de configuración.

Al proteger las contraseñas, debe utilizar una clave de cifrado fuerte y personalizada. Para acceder a las contraseñas durante el tiempo de ejecución, esta clave de cifrado se debe proporcionar a AMS.

Existen dos métodos para proporcionar la ubicación del archivo de claves de cifrado, que son, a través de:

- Propiedad de configuración **amscred.keyfile** en el archivo `keystore.conf`
- Variable de entorno de **MQS_AMSCRED_KEYFILE**

El orden de prioridad es **MQS_AMSCRED_KEYFILE**, seguido de **amscred.keyfile**y, a continuación, la clave predeterminada.

Conceptos relacionados

“Nombres distinguidos de emisor en AMS” en la página 683
Los nombres distinguidos de emisor identifican usuarios que están autorizados a colocar mensajes en una cola. Un remitente utiliza su certificado para firmar un mensaje, antes de colocar el mensaje en una cola.

“Nombres distinguidos de destinatario en AMS” en la página 685
Los nombres distinguidos de destinatario identifican a usuarios que están autorizados a recuperar mensajes de una cola.

Estructura del archivo de configuración del almacén de claves (keystore.conf) para AMS

El archivo de configuración del almacén de claves (`keystore.conf`) apunta Advanced Message Security a la ubicación del almacén de claves adecuado.

Cada uno de los siguientes tipos de archivo de configuración tiene un prefijo:

AMSCRED
Parámetros relacionados con el sistema de protección de contraseñas.

CMS
Certificate Management System, las entradas de configuración tienen el prefijo: `cms.`

PKCS#11
Public Key Cryptography Standard #11, las entradas de configuración tienen el prefijo: `pkcs11.`

IBM i **PEM**
Formato Privacy Enhanced Mail, las entradas de configuración tienen el prefijo: `pem.`

JKS
Java KeyStore, las entradas de configuración tienen el prefijo: `jks.`

JCEKS
Java Cryptographic Encryption KeyStore, las entradas de configuración tienen el prefijo: `jceks.`

z/OS MQ Adv. VUE **JCERACFKS**
Java Cryptographic Encryption RACF keyring KeyStore, las entradas de configuración tienen el prefijo: `jceracfks`.

Importante: A partir de IBM MQ 9.0, no se tienen en cuenta los valores de `JCEKS.provider` y `JKS.provider`. Se utiliza el proveedor Bouncy Castle, junto con el suministro JCE/JCE que proporcione el JRE en uso. Para obtener más información, consulte “Soporte para JRE que no son deIBM con AMS” en la página 660.

Estructuras de ejemplo para almacenes de claves:

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
```

Java JKS

```
jks.keystore = dir/Keystore
jks.certificate = certificate_label
jks.encrypted = no
jks.keystore_pass = password
jks.key_pass = password
```

Java JCEKS

```
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
```

Tabla 104. Resumen de los parámetros necesarios para cada tipo de archivo de configuración

Parámetros	Obligatorio	Tipo de archivo de configuración				
		Java (PKCS#11, JKS, JCEKS y 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	✓	✓		✓		

Tabla 104. Resumen de los parámetros necesarios para cada tipo de archivo de configuración (continuación)

Parámetros	Obligatorio	Tipo de archivo de configuración				
		Java (PKCS#11, JKS, JCEKS y JCERACFKS)	IBM i PEM	PKCS#11	CMS	AMSCRED
`token_pin`	✓	✓		✓		
`secondary_keystore`	✓	✓		✓		
`secondary_keystore_password`	✓	✓				
`encrypted`		✓	IBM i ✓	✓		
`keystore_pass`	✓	✓				
`key_pass`		✓				
`provider`		✓				
`keyfile`						✓ Usted

Tenga en cuenta que puede añadir comentarios utilizando el símbolo # .

Los parámetros del archivo de configuración se definen del modo siguiente:

keystore
Sólo configuración de CMS y Java.

Vía de acceso al archivo de claves para la configuración de CMS, JKS y JCEKS.

z/OS MQ Adv. VUE URI al archivo de claves RACF para la configuración de JCERACFKS.

Importante:

- La vía de acceso del archivo de almacén de claves no debe incluir la extensión de archivo.
- z/OS MQ Adv. VUE El URI al archivo de claves RACF debe tener el formato:

  ```
  safkeyring://user/keyring
  ```

  donde:

  - *user* es el ID de usuario propietario del conjunto de claves
  - *keyring* es el nombre del conjunto de claves.

IBM i **private**
Sólo configuración de PEM.

Nombre de un archivo que contiene la clave privada y el certificado en formato PEM.

IBM i **public**
Sólo configuración de PEM.

Nombre de un archivo que contiene certificados públicos de confianza en formato PEM.

IBM i **password**

Sólo configuración de PEM.

Contraseña que se utiliza para descifrar una clave privada cifrada.

Debe proteger este campo utilizando la herramienta de protección de contraseña nativa de AMS ; consulte “Protección de contraseñas” en la página 660

library

Sólo PKCS#11.

Nombre de vía de acceso de la biblioteca PKCS#11.

certificate

Sólo configuración de CMS, PKCS#11 y Java.

Etiqueta del certificado.

token

Sólo PKCS#11.

Etiqueta de señal.

token_pin

Sólo PKCS#11.

PIN para desbloquear la señal.

Sólo para operaciones de Java ; debe proteger este campo utilizando la herramienta de protección por contraseña de Java AMS ; consulte “Protección de contraseñas” en la página 660.

Sólo para operaciones nativas; debe proteger este campo utilizando la herramienta de protección por contraseña de AMS nativa; consulte “Protección de contraseñas” en la página 660.

secondary_keystore

Sólo PKCS#11.

Nombre de vía de acceso del almacén de claves CMS, que se proporciona sin la extensión `.kdb`, que contiene certificados de ancla (certificados raíz) necesarios para los certificados almacenados en la señal PKCS #11. El almacén de claves secundario también puede contener certificados que se intermedios en la cadena de confianza, así como certificados de destinatario que se definen en la política de seguridad de privacidad. Este almacén de claves CMS debe ir acompañado de un archivo de ocultación que debe estar ubicado en el mismo directorio que el almacén de claves secundario.

Para entornos Java es necesario un almacén de claves JKS y debe proporcionar un **secondary_keystore_password**.

secondary_keystore_password

Java Sólo PKCS#11.

Contraseña para el almacén de claves JKS proporcionado a través de la propiedad `secondary_keystore` . Debe proteger este campo utilizando la herramienta de protección de contraseñas de Java AMS ; consulte “Protección de contraseñas” en la página 660.

encrypted

Java y, desde IBM MQ 9.3.0, PKCS#11 y IBM i PEM únicamente.

Estado de la contraseña.

keystore_pass

Sólo configuración de Java.

Contraseña del archivo de almacén de claves.

Sólo para operaciones Java . Debe proteger este campo utilizando la herramienta de protección de contraseñas de Java AMS ; consulte “Protección de contraseñas” en la página 660.

key_pass

Sólo configuración de Java.

Contraseña para la clave privada del usuario.

Sólo para operaciones de Java ; debe proteger este campo utilizando la herramienta de protección por contraseña de Java AMS ; consulte “Protección de contraseñas” en la página 660.

keyfile
Proporciona la ubicación de la clave inicial que se debe utilizar al proteger o descifrar contraseñas contenidas en este archivo de configuración; consulte “Protección de contraseñas” en la página 660

provider
Sólo configuración de Java.

Proveedor de seguridad de Java que aplica los algoritmos criptográficos necesarios para el certificado de almacén de claves.

Importante: la información almacenada en el almacén de claves es esencial para el flujo seguro de los datos enviados utilizando IBM MQ. Los administradores de seguridad deben prestar especial atención al asignar permisos de archivo para estos archivos.

Protección de contraseñas

Debe proteger las contraseñas y otra información confidencial contenida en el archivo `keystore.conf` . Para obtener más información, consulte **runamscred**.

Ejemplo del archivo `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
```

Tareas relacionadas

“Configuración de la protección por contraseña de AMS para los archivos de configuración” en la página 674

El almacenamiento de contraseñas de claves privadas y de almacén de claves como texto sin formato supone un riesgo para la seguridad, por lo que Advanced Message Security proporciona una herramienta que puede codificar esas contraseñas utilizando la clave de un usuario.

Soporte para JRE que no son deIBM con AMS

IBM MQ classes for Java y IBM MQ classes for JMS dan soporte a la operación de Advanced Message Security cuando se ejecutan con JRE no IBM.

Advanced Message Security (AMS) implementa Cryptographic Message Syntax (CMS). La sintaxis CMS se utiliza firmar digitalmente, resumir, autenticar o cifrar contenido arbitrario de mensajes.

Desde IBM MQ 9.0, el soporte de Advanced Message Security en IBM MQ classes for Java y IBM MQ classes for JMS utiliza los paquetes Bouncy Castle de código abierto para dar soporte a CMS. Esto significa que estas clases pueden dar soporte a la operación de Advanced Message Security cuando se ejecutan con JRE que no son IBM.

Antes de IBM MQ 9.0, Advanced Message Security no estaba soportado en JRE que no son IBM en clientes Java. El soporte de Advanced Message Security en IBM MQ classes for Java y IBM MQ classes for JMS dependía del soporte de CMS proporcionado específicamente por la implementación de IBM de Java Cryptography Extensions (JCE). Debido a esta restricción, la funcionalidad solo estaba disponible cuando se utilizaba un Java runtime environment (JRE) que incluía el proveedor JCE de Java.

Ubicación y numeración de versiones para archivos JAR de Bouncy Castle

Los archivos JAR de Bouncy Castle que son necesarios para el soporte para los JRE que no son IBM se incluyen como parte del paquete de instalación de IBM MQ classes for Java y IBM MQ classes for JMS.

Los archivos JAR de Bouncy Castle utilizados son los archivos siguientes:

El archivo JAR proporcionado, que es básico para las operaciones de Bouncy Castle.

V 9.4.0 A partir de IBM MQ 9.4.0, este archivo JAR se denomina `bcprov-jdk18on.jar`.

El archivo JAR "PKIX", que contiene el soporte para las operaciones CMS que utiliza Advanced Message Security.

V 9.4.0 A partir de IBM MQ 9.4.0, este archivo JAR se denomina `bcpkix-jdk18on.jar`.

El archivo JAR "util", que contiene las clases utilizadas por los otros archivos JAR de Bouncy Castle.

V 9.4.0 A partir de IBM MQ 9.4.0, este archivo JAR se denomina `bcutil-jdk18on.jar`.

Dependencias

Las clases de IBM MQ 9.1 y clases posteriores se han probado con IBM JRE y Oracle JRE. También es probable que ejecuten correctamente en cualquier JRE compatible con J2SE. Sin embargo, debe tener en cuenta las dependencias siguientes:

- No hay ningún cambio en la configuración de Advanced Message Security.
- Las clases Bouncy Castle se utilizan solo para operaciones de CMS. Todas las demás operaciones relacionadas con la seguridad, por ejemplo, el acceso al almacén de claves, el cifrado real de datos y el cálculo de sumas de comprobación de firma, utilizan la funcionalidad proporcionada por el JRE.

  Importante: Por este motivo, el JRE utilizado debe incluir una implementación del proveedor de JCE.
- Para utilizar algunos algoritmos de cifrado de *alta seguridad*, es posible que tenga que instalar los archivos de política *no restringidos* para la implementación JCE del JRE.

  Consulte la documentación de JRE para obtener más detalles.
- Si ha habilitado la seguridad de Java:
  - Añada `java.security.SecurityPermissioninsertProvider.BC` a la aplicación para que las clases Bouncy Castle se puedan utilizar como proveedor de seguridad.
  - Otorgue `java.security.AllPermission` a los archivos JAR de Bouncy Castle.

    V 9.4.0 A partir de IBM MQ 9.4.0, estos archivos son:

    mq_install_dir`/java/lib/bcutil-jdk18on.jar`
    mq_install_dir`/java/lib/bcpkix-jdk18on.jar`
    mq_install_dir`/java/lib/bcprov-jdk18on.jar`

Conceptos relacionados

Qué se instala para las clases de IBM MQ para JMS
Qué se instala para las clases de IBM MQ para Java

Multi Intercepción del agente de canal de mensajes (MCA) y AMS

La intercepción MCA permite a un gestor de colas que se ejecuta bajo IBM MQ habilitar de forma selectiva políticas que se van a aplicar para canales de conexión de servidor.

La interceptación de MCA permite a los clientes que permanecen fuera de AMS seguir conectados a un gestor de colas y cifrar y descifrar sus mensajes.

La intercepción MCA se ha diseñado para proporcionar la prestación AMS cuando AMS no se puede habilitar en el cliente. Tenga en cuenta que utilizar la intercepción MCA y un cliente habilitado para AMS lleva a una doble protección que podría ser problemática para recibir aplicaciones. Para obtener más información, consulte “Inhabilitación de Advanced Message Security en el cliente” en la página 664.

Nota: Los interceptores MCA no están soportados para los canales AMQP o MQTT.

Archivo de configuración del almacén de claves

De forma predeterminada, el archivo de configuración de almacén de claves para la interceptación de MCA es `keystore.conf` y se encuentra en el directorio `.mqs` de la vía de acceso del directorio HOME inicial del usuario que ha iniciado el gestor de colas o el escucha. El almacén de claves también se puede utilizar mediante la variable de entorno MQS_KEYSTORE_CONF. Si desea más información sobre cómo configurar el almacén de claves de AMS, consulte "Utilización de almacenes de claves y certificados con AMS" en la página 655.

Para habilitar la interceptación de MCA, debe proporcionar el nombre de un canal que desee utilizar en el archivo de configuración de almacén de claves. Para la intercepción MCA, solo se puede utilizar un tipo de almacén de claves cms.

Consulte "Ejemplo de intercepción de MCA para AMS" en la página 662 si desea un ejemplo de configuración de la intercepción MCA.

Atención: Debe completar la autenticación de cliente y el cifrado en los canales seleccionados, por ejemplo, utilizando SSL y SSLPEER o CHLAUTH TYPE(SSLPEERMAP), para asegurarse de que solo los clientes autorizados se pueden conectar a y utilizar esta prestación.

IBM i

Si su empresa utiliza IBM iy ha seleccionado una entidad emisora de certificados (CA) comercial para firmar el certificado, el Certificate Manager digital crea una solicitud de certificado en formato PEM (Privacy-Enhanced Mail). Debe enviar la solicitud a su CA elegida.

Para ello, debe utilizar el mandato siguiente para seleccionar el certificado correcto para el canal especificado en `channelname`:

```
pem.certificate.channel.channelname
```

Ejemplo de intercepción de MCA para AMS

Una tarea de ejemplo de cómo configurar una interceptación de MCA de AMS.

Antes de empezar

Atención: Debe completar la autenticación de cliente y el cifrado en los canales seleccionados, por ejemplo, utilizando SSL y SSLPEER o CHLAUTH TYPE(SSLPEERMAP), para asegurarse de que solo los clientes autorizados se pueden conectar a y utilizar esta prestación.

Si su empresa utiliza IBM iy ha seleccionado una entidad emisora de certificados (CA) comercial para firmar el certificado, el Certificate Manager digital crea una solicitud de certificado en formato PEM (Privacy-Enhanced Mail). Debe enviar la solicitud a su CA elegida.

Acerca de esta tarea

Esta tarea le guía a través del proceso de configuración del sistema para utilizar la intercepción MCA y, después, de la verificación de la configuración.

Nota: IBM MQ, incluye los interceptores de AMS y los habilita dinámicamente en los entornos de ejecución de cliente y servidor de MQ .

Atención:

- Sustituya `userID` en el código por su ID de usuario.
- El procedimiento siguiente no funciona como se esperaba en IBM MQ a menos que la intercepción AMS esté desactivada en el cliente.

Procedimiento

1. Cree la base de datos de claves y los certificados utilizando los siguientes mandatos para crear un script de shell-

   Además, cambie **INSTLOC** y **KEYSTORELOC** o ejecute los mandatos necesarios. Tenga en cuenta que podría no necesitar crear el certificado para 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. Comparta los certificados entre las dos bases de datos para que cada usuario pueda identificar correctamente al otro.

   Es importante que utilice el método descrito para compartir certificados en la *Guía de inicio rápido*, para la plataforma que utiliza la empresa:

   Windows
   Tarea 5 Compartir certificados

   AIX and Linux
   Tarea 5 Compartir certificados

   Java clientes
   Tarea 5 Compartir certificados

3. Cree `keystore.conf` con la siguiente configuración: `Keystore.conf location: /home/userID/ssl/ams1/`

```
cms.keystore = /home/userID/ssl/ams1/alicekey
cms.certificate.channel.SYSTEM.DEF.SVRCONN = alice_cert
```

   Atención:

   a. El almacén de claves debe estar en el sistema donde está el gestor de colas.

   b. Debe especificar un canal específico para `cms.certificate` para habilitar la intervención de MCA y, a continuación, el gestor de colas realiza operaciones AMS en aplicaciones que se conectan a través de ese canal a colas con políticas establecidas.

4. Cree e inicie el gestor de colas `AMSQMGR1`
5. Defina un escucha TCP utilizando un número de puerto disponible bajo el control QMGR.

   Por ejemplo:

```
DEFINE LISTENER(MY.LISTENER) TRPTYPE(TCP) PORT(14567) CONTROL(QMGR)
```

6. Inicie el escucha y verifique que se ha iniciado correctamente.

   Por ejemplo:

```
START  LISTENER(MY.LISTENER)
DISPLAY LSSTATUS(MY.LISTENER) PORT
```

7. Detenga el gestor de colas.
8. Establezca el almacén de claves:

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. Inicie el gestor de colas en el mismo shell, de forma que la variable de entorno MQS_KEYSTORE_CONF esté disponible para el gestor de colas.
10. Establezca la política de seguridad y 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 y dspmqspl si desea más información.
11. Establezca la variable de entorno *MQSERVER* :

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. Elimine la política de seguridad y verifique el resultado:

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

13. Examine la cola desde la instalación de IBM MQ 9.4:

```
/opt/mq93/samp/bin/amqsbcg TESTQ AMSQMGR1
```

    El resultado muestra los mensajes en formato cifrado.
14. Establezca la política de seguridad y verifique el 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. Ejecute **amqsgetc** desde la instalación de IBM MQ 9.4:

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

Conceptos relacionados

“Estructura del archivo de configuración del almacén de claves (keystore.conf) para AMS” en la página 656

El archivo de configuración del almacén de claves (`keystore.conf`) apunta Advanced Message Security a la ubicación del almacén de claves adecuado.

Referencia relacionada

“Limitaciones conocidas de AMS” en la página 614

Existe un número de opciones de IBM MQ que no están soportadas o que tienen limitaciones para Advanced Message Security.

Inhabilitación de Advanced Message Security en el cliente

Es necesario inhabilitar IBM MQ Advanced Message Security (AMS) si está utilizando un cliente IBM MQ para conectarse a un gestor de colas desde una versión anterior del producto y se notifica un error 2085 (`MQRC_UNKNOWN_OBJECT_NAME`) .

Acerca de esta tarea

IBM MQ Advanced Message Security (AMS) se habilita automáticamente en un cliente IBM MQ y, por lo tanto, de forma predeterminada, el cliente intenta comprobar las políticas de seguridad para los objetos en el gestor de colas.

Si se notifica este error, cuando intenta conectarse a un gestor de colas desde una versión anterior del producto, puede inhabilitar AMS del modo siguiente:

- Para clientes de Java, en cualquiera de las formas siguientes:
  - Estableciendo una variable de entorno **AMQ_DISABLE_CLIENT_AMS**.
  - Estableciendo la propiedad del sistema Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS.

  - Utilizando la propiedad **DisableClientAMS** , en la stanza Security del archivo `mqclient.ini` .
- Para clientes C, estableciendo una variable de entorno **MQS_DISABLE_ALL_INTERCEPT**.

Nota: No puede utilizar la variable de entorno **AMQ_DISABLE_CLIENT_AMS** para clientes C. En su lugar, debe utilizar la variable de entorno **MQS_DISABLE_ALL_INTERCEPT** .

Procedimiento

- Para inhabilitar AMS en el cliente, utilice una de las opciones siguientes:

  Variable de entorno AMQ_DISABLE_CLIENT_AMS
  Es necesario establecer esta variable en los siguientes casos:

  - Si utiliza un Java runtime environment (JRE) que no sea el IBM Java runtime environment (JRE)
  - Si está utilizando un cliente IBM MQ IBM MQ classes for JMS o IBM MQ classes for Java .

  Cree la variable de entorno **AMQ_DISABLE_CLIENT_AMS** y establézcala en TRUE en el entorno donde se ejecuta la aplicación. Por ejemplo:

  ```
  export AMQ_DISABLE_CLIENT_AMS=TRUE
  ```

  La propiedad de sistema Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS
  Para clientes IBM MQ classes for JMS y IBM MQ classes for Java, puede establecer la propiedad del sistema Java com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS en el valor TRUE para la aplicación Java.

  Por ejemplo, puede establecer la propiedad del sistema Java como una opción `-D` cuando se invoca el mandato Java:

  ```
  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
  ```

  De forma alternativa, puede especificar la propiedad del sistema Java dentro de un archivo de configuración de JMS, `jms.config`, si la aplicación utiliza este archivo.

  Variable de entorno MQS_DISABLE_ALL_INTERCEPT
  Debe establecer esta variable de entorno si utiliza IBM MQ con clientes nativos y necesita inhabilitar AMS en el cliente.

  Cree la variable de entorno **MQS_DISABLE_ALL_INTERCEPT** y establézcala en TRUE en el entorno donde se ejecuta el cliente. Por ejemplo:

  ```
  export MQS_DISABLE_ALL_INTERCEPT =TRUE
  ```

  Puede utilizar la variable de entorno **MQS_DISABLE_ALL_INTERCEPT** sólo para clientes C. Para los clientes Java , en su lugar debe utilizar la variable de entorno **AMQ_DISABLE_CLIENT_AMS** .

  Propiedad DisableClientAMS en el archivo mqclient.ini
  Puede utilizar esta opción para clientes IBM MQ classes for JMS y IBM MQ classes for Java y para clientes C.

  Añada el nombre de propiedad DisableClientAMS bajo la stanza **Security** en el archivo `mqclient.ini`, tal como se muestra en el ejemplo siguiente:

  ```
  Security:
  DisableClientAMS=Yes
  ```

  También puede habilitar AMS tal como se indica en el ejemplo siguiente:

  ```
  Security:
  DisableClientAMS=No
  ```

Qué hacer a continuación

Para obtener más información sobre los problemas con la apertura de colas protegidas de AMS , consulte Problemas con la apertura de colas protegidas cuando se utiliza AMS con JMS.

Conceptos relacionados

"Intercepción del agente de canal de mensajes (MCA) y AMS" en la página 661
La intercepción MCA permite a un gestor de colas que se ejecuta bajo IBM MQ habilitar de forma selectiva políticas que se van a aplicar para canales de conexión de servidor.

Tareas relacionadas

Archivo de configuración de IBM MQ MQI client , `mqclient.ini`

Referencia relacionada

El archivo de configuración IBM MQ classes for JMS

Requisitos de certificado para AMS

Los certificados deben tener una clave pública RSA para utilizarlos con Advanced Message Security.

Para obtener más información sobre distintos tipos de clave pública y cómo crearlos, consulte "Certificados digitales y compatibilidad de CipherSpec en IBM MQ" en la página 49.

Extensiones de uso de claves

Las extensiones de uso de claves limitan adicionalmente la forma en la que un certificado se puede utilizar.

En Advanced Message Security, el uso de claves de los certificados X.509 v3 se debe establecer de acuerdo con la especificación RFC 5280.

Para la calidad de la integridad de protección, si se establecen las extensiones de uso de clave de certificado, dicho conjunto debe incluir al menos uno de los dos:

- **`nonRepudiation`**
- **`digitalSignature`**

Para la calidad de la integridad de protección, si se establecen las extensiones de uso de clave de certificado, dicho conjunto debe incluir:

- **`keyEncipherment`**

Para la confidencialidad de la calidad de protección, si se establecen las extensiones de uso de clave de certificado, dicho conjunto debe incluir:

- **`dataEncipherment`**

El uso de claves ampliado limita aún más las extensiones de uso de claves. Para todas las calidades de protección, si se establece el uso de claves ampliado del certificado, el conjunto debe incluir:

- **`emailProtection`**

Conceptos relacionados

"Calidad de protección en AMS" en la página 686
Las políticas de protección de datos de Advanced Message Security implican una calidad de protección (QOP).

Métodos de validación de certificados en AMS

Puede utilizar Advanced Message Security para detectar y rechazar los certificados revocados para que los mensajes de las colas no estén protegidos mediante certificados que no cumplen los estándares de seguridad.

AMS le permite verificar la validez de un certificado utilizando Online Certificate Status Protocol (OCSP) o la lista de revocación de certificados (CRL).

AMS se puede configurar para realizar la comprobación mediante OCSP o CRL, o por ambos métodos. Si se habilitan ambos métodos, entonces AMS utiliza primero OCSP para el estado de revocación por motivos de rendimiento. Si el estado de revocación de un certificado es indeterminado después de la comprobación por OCSP, AMS utiliza la comprobación por CRL.

Tenga en cuenta que tanto la comprobación OCSP como la comprobación CRL están habilitadas de forma predeterminada.

Conceptos relacionados

"Protocolo de estado de certificado en línea (OCSP) en AMS" en la página 667
El protocolo de estado de certificado en línea (OCSP) determina si un certificado se ha revocado y, por consiguiente, ayuda a determinar si el certificado puede ser de confianza. De forma predeterminada OCSP está habilitado.

"Listas de revocación de certificados (CRL) en AMS" en la página 669
Las CRL contienen una lista de certificados que han sido marcados por la Autoridad de certificación (CA) como no fiables por diversas razones, por ejemplo, porque la clave privada se ha perdido o está en peligro.

Protocolo de estado de certificado en línea (OCSP) en AMS

El protocolo de estado de certificado en línea (OCSP) determina si un certificado se ha revocado y, por consiguiente, ayuda a determinar si el certificado puede ser de confianza. De forma predeterminada OCSP está habilitado.

OCSP no está soportado en sistemas IBM i.

Habilitación de la comprobación de OCSP para interceptores nativos de Advanced Message Security
La comprobación de OCSP (Online Certificate Status Protocol) en Advanced Message Security está habilitada de forma determinada, según la información de los certificados que se utilizan.

Procedimiento

Añada las opciones siguientes al archivo de configuración del almacén de claves:

Nota: Todo lo incluido en la stanza OCSP es opcional y puede especificarse de forma independiente.

Opción	Descripción
`ocsp.enable=off`	Habilite la comprobación de OCSP si el certificado que se está comprobando tiene una extensión AIA (Authority Info Access) con un método de acceso PKIX_AD_OCSP que contiene un URI de apunta a la ubicación del respondedor de OCSP. Valores posibles: on u `off`.
`ocsp.url=`*`responder_URL`*	Dirección de URL del respondedor de OCSP. Si se omite esta opción, la comprobación de OCSP no AIA se inhabilita.
`ocsp.http.proxy.host=`*`OCSP_proxy`*	Dirección de URL del servidor proxy de OCSP. Si se omite esta opción, no se utiliza un proxy para las comprobaciones de certificado en línea no de AIA.
`ocsp.http.proxy.port=`*`port_number`*	Número de puerto del servidor proxy de OCSP. Si se omite esta opción, se utiliza el puerto predeterminado 8080.
`ocsp.nonce.generation=on/off`	Generar valor de seguridad al consultar OCSP. El valor predeterminado es `off`.

Opción	Descripción
`ocsp.nonce.check=on/off`	Comprobar valor de seguridad después de recibir una respuesta de OCSP. El valor predeterminado es `off`.
`ocsp.nonce.size=8`	Tamaño del valor de seguridad, en bytes.
`ocsp.http.get=on/off`	Especificar HTTP GET como método de solicitud. Si esta opción se establece en `off`, se utiliza HTTP POST. El valor predeterminado es `off`.
`ocsp.max_response_size=20480`	Tamaño máximo de la respuesta proporcionada por el programa de respuesta de OCSP, en bytes.
`ocsp.cache_size=100`	Habilitar el almacenamiento en memoria caché interna de la respuesta de OCSP y establecer el número límite de entradas de la memoria caché.
`ocsp.timeout=30`	Tiempo de espera para la respuesta de un servidor, en segundos, pasado el cual Advanced Message Security concluye.
`ocsp.unknown=ACCEPT`	Defina el comportamiento cuando un servidor OCSP no se puede alcanzar dentro de un periodo de tiempo de espera. Valores posibles: • `ACCEPT` Permite el certificado • `WARN` Permite el certificado y registra un aviso • `REJECT` Impide que el certificado se use y anota un error

Habilitación de la comprobación de OCSP en Java en AMS
Para habilitar la comprobación de OCSP para Java en Advanced Message Security, modifique el archivo `java.security` o el archivo de configuración del almacén de claves.

Acerca de esta tarea

Existen dos formas de habilitar la comprobación de OCSP en Advanced Message Security:

Utilización de java.security
Compruebe si el certificado contiene una extensión de certificado AIS (Authority Information Access).

Procedimiento

1. Si AIA no está configurado o desea alterar el certificado, edite el archivo `$JAVA_HOME/lib/security/java.security` con las propiedades siguientes:

   ```
   ocsp.responderURL=http://url.to.responder:port
   ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
   ```

   y habilite la comprobación de OCSP editando el archivo `$JAVA_HOME/lib/security/java.security` con la línea siguiente:

   ```
   ocsp.enable=true
   ```

2. Si AIA está configurado, habilite la comprobación de OCSP editando el archivo `$JAVA_HOME/lib/security/java.security` con la línea siguiente:

```
ocsp.enable=true
```

Qué hacer a continuación

Si está utilizando Java Security Manager, para completar la configuración, añada el permiso siguiente de Java a `lib/security/java.policy`

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

Utilización de keystore.conf

Procedimiento

Añada el atributo siguiente al archivo de configuración:

```
ocsp.enable=true
```

Importante: Cuando este atributo está definido en el archivo de configuración, prevalece sobre los valores contenidos en java.security.

Qué hacer a continuación

Para completar la configuración, añada los permisos siguientes de Java a `lib/security/java.policy`:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

Listas de revocación de certificados (CRL) en AMS

Las CRL contienen una lista de certificados que han sido marcados por la Autoridad de certificación (CA) como no fiables por diversas razones, por ejemplo, porque la clave privada se ha perdido o está en peligro.

Para validar certificados, Advanced Message Security crea una cadena de certificado que consta del certificado del firmante y el certificado de la entidad emisora de certificados hasta llegar a un ancla de confianza. Un ancla de confianza es un archivo de almacén de confianza que contiene un certificado de confianza o un certificado raíz de confianza que se utiliza para confirmar la confianza de un certificado. AMS verifica la vía de acceso del certificado utilizando un algoritmo de validación PKIX. Cuando se crea y verifica la cadena, AMS completa la validación de certificados, que incluye validar la fecha de emisión y caducidad de cada certificado de la cadena por comparación con la fecha actual y, comprobar si la extensión de uso de la clave está presente en el certificado de entidad final. Si la extensión se añade al certificado, AMS verifica si **digitalSignature** o **nonRepudiation** también están definidos. Si no lo están, se notifica y registra un error de seguridad `MQRC_SECURITY_ERROR`. A continuación, AMS descarga listas de revocación de certificados a partir de archivos o de LDAP, dependiendo de los valores que se hayan especificado en el archivo de configuración. AMS sólo es compatible con las listas de revocación de certificados que estén codificadas en el formato DER. Si no se encuentra ninguna configuración de CRL en el archivo de configuración del almacén de claves, AMS no realiza ninguna comprobación de validez por CRL. Para cada certificado de entidad emisora de certificados (certificados de CA), AMS consulta a LDAP para conocer las listas de revocación de certificados (CRL) utilizando nombres distinguidos de una CA para encontrar su CRL. La consulta a LDAP incluye los atributos siguientes:

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
deltaRevocationList
deltaRevocationList;binary,
```

Nota: `deltaRevocationList` sólo se puede utilizar cuando se especifica como puntos de distribución.

Habilitación de la validación de certificados y del soporte de lista de revocación de certificados (CRL) en interceptores nativos
Debe modificar el archivo de configuración del almacén de claves de manera que Advanced Message Security pueda descargar las CRL del servidor LDAP (Lightweight Directory Access Protocol).

Acerca de esta tarea

IBM i La habilitación del soporte de lista de revocación de certificados y de la validación de certificados en los interceptores nativos no está soportado para Advanced Message Security en IBM i.

Procedimiento

Añada las opciones siguientes al archivo de configuración:

Nota: Todo lo incluido en la stanza CRL es opcional y puede especificarse de forma independiente.

Opción	Descripción
`crl.ldap.host=`*`host_name`*	Nombre de host del servidor LDAP.
`crl.ldap.port=`*`port_number`*	Número de puerto del servidor LDAP. Puede especificar un máximo de 11 servidores. Se utilizan varios hosts LDAP para asegurar un relevo transparente en caso de que falle la conexión LDAP. Todos los servidores LDAP son réplicas y contienen los mismos datos. Cuando el interceptor de AMS Java se conecta satisfactoriamente a un servidor LDAP, no intenta descargar CRL de los servidores proporcionados restantes.
`crl.cdp=off`	Utilice esta opción para comprobar o utilizar extensiones CRLDistributionPoints en certificados.
`crl.ldap.version=3`	Número de versión del protocolo LDAP. Valores posibles: 2 ó 3.
`crl.ldap.user=cn=`*`username`*	Inicio de sesión en el servidor LDAP. Si no se especifica este valor, los atributos de CRL en LDAP deben poder ser leídos por todos los usuarios.
`crl.ldap.pass=`*`password`*	Contraseña del servidor LDAP.
`crl.ldap.encrypted=`*`no/yes`*	Indica si el `clr.ldap.pass` está cifrado o no. Consulte Protección de contraseñas en archivos de configuración de AMS para obtener más información.
`crl.ldap.cache_lifetime=0`	Tiempo de vida de la memoria caché de LDAP, expresado en segundos. Valores posibles: 0-86400.
`crl.ldap.cache_size=50`	Tamaño de la memoria caché de LDAP. Esta opción se puede especificar sólo si el valor de `crl.ldap.cache_lifetime` es mayor que `0`.
`crl.http.proxy.host=some.host.com`	Puerto del servidor proxy HTTP para recuperar CRL CDP.
`crl.http.proxy.port=8080`	Número de puerto del servidor proxy HTTP.

Opción	Descripción
`crl.http.max_response_size=204800`	El tamaño máximo de CRL, en bytes, que se puede recuperar de un servidor HTTP aceptado por IBM Global Security Kit (GSKit).
`crl.http.timeout=30`	Tiempo de espera para la respuesta de un servidor, en segundos, pasado el cual AMS concluye.
`crl.http.cache_size=0`	Tamaño de la memoria caché de HTTP, en bytes.
`crl.unknown=ACCEPT`	Defina el comportamiento cuando un servidor CRL no se puede alcanzar dentro de un periodo de tiempo de espera. Valores posibles: • `ACCEPT` Permite el certificado • `WARN` Permite el certificado y registra un aviso • `REJECT` Impide que el certificado se use y anota un error

Habilitación del soporte de las listas de revocación de certificados en Java in AMS
Para habilitar las listas de revocación de certificados en Advanced Message Security, debe modificar el archivo de configuración del almacén de claves para permitir que AMS descargue las CRL desde el servidor LDAP (Lightweight Directory Access Protocol) y configurar el archivo java.security.

Procedimiento

1. Añada las opciones siguientes al archivo de configuración:

Cabecera	Descripción
`crl.ldap.host=`*`host_name`*	Nombre de host de LDAP.
`crl.ldap.port=`*`port_number`*	Número de puerto del servidor LDAP. Puede especificar un máximo de 11 servidores. Se utilizan varios hosts LDAP para asegurar un relevo transparente en caso de que falle la conexión LDAP. Todos los servidores LDAP son réplicas y contienen los mismos datos. Cuando el interceptor de AMS Java se conecta satisfactoriamente a un servidor LDAP, no intenta descargar CRL de los servidores proporcionados restantes. Java no utiliza los valores `crl.ldap.user` y crl.ldaworldp.pass. Java no utiliza un usuario y una contraseña cuando se conecta a un servidor LDAP. Como consecuencia, los atributos de CRL en LDAP deben poder ser leídos por todos los usuarios.
`crl.cdp=on/off`	Utilice esta opción para comprobar o utilizar extensiones CRLDistributionPoints en certificados.

2. Modifique el archivo `JRE/lib/security/java.security` con las propiedades siguientes:

Nombre de propiedad	Descripción
`com.ibm.security.enableCRLDP`	Esta propiedad toma los valores siguientes: `true`, `false`. Si se establece en `true`, cuando se realiza comprobación de revocación de certificados, las CRL se localizan utilizando el URL de la extensión de puntos de distribución del certificado. Si se establece en `false` o no se establece, se inhabilita la comprobación de CRL mediante la extensión de puntos de distribución de CRL.
`ibm.security.certpath.ldap.cache.lifetime`	Utilice esta propiedad para establecer un valor en segundos para el tiempo de vida de las entradas de la memoria caché del almacén de certificados de LDAP. El valor 0 inhabilita la memoria caché; -1 significa un tipo de vida ilimitado. Si no se define un valor, el tiempo de vida predeterminado es 30 segundos.
`com.ibm.security.enableAIAEXT`	Esta propiedad toma los valores siguientes: `true`, `false`. Si la propiedad se establece en `true`, se examina cualquier extensión de AIA (Authority Information Access) que se encuentre en la vía de acceso del certificados para determinar si contiene los URI de LDAP. Para cada URI de LDAP encontrado, se crea un objeto LDAPCertStore y se añade a la colección de almacenes de certificados que se utiliza para localizar otros certificados que son necesarios para crear la vía de acceso del certificado. Si la propiedad se establece en `false` o no se define, no se crean más objetos LDAPCertStore.

z/OS *Habilitación de las listas de revocación de certificados (CRL) en z/OS*

Advanced Message Security da soporte a la lista de revocación de certificados (CRL) y comprueba los certificados digitales que se utilizan para proteger los mensajes de datos.

Acerca de esta tarea

Cuando está habilitado, Advanced Message Security validará los certificados de destinatarios cuando se colocan mensajes en una cola con protección de privacidad y validarán los certificados de emisor cuando se recuperen los mensajes desde una cola con protección (de integridad o privacidad). En este caso la validación incluye verificar que no se registren certificados relevantes en un CRL relevante.

Advanced Message Security utiliza los servicios de IBM System SSL para validar los certificados de emisores y destinatarios. Puede encontrar documentación detallada sobre la validación de certificados SSL del sistema en el manual z/OS Cryptographic Services System Secure Sockets Layer Programming .

Para habilitar la comprobación CRL, especifique la ubicación de un archivo de configuración CRL mediante CRLFILE DD en el JCL de la tarea inicial para el espacio de direcciones AMS. Se proporciona un archivo de configuración CRL de ejemplo en *thlqual*.SCSQPROC(CSQ40CRL). Los valores permitidos en este archivo son los siguientes:

Tabla 105. Variables de configuración CRL de Advanced Message Security		
Variable	**Valores válidos**	**Descripción**
crl.ldap.host[.n]	*nombrehost -o- nombrehost:puerto*	La direcciónip/nombrehost de su servidor LDAP que aloja los CRL de sus certificados de emisor. Si no especifica un número para su servidor LDAP, se utiliza el número de puerto especificado mediante crl.ldap.port.
crl.ldap.port	*puerto*	El número de puerto TCP/IP de su servidor LDAP.
crl.ldap.user	*usuario_ldap*	El nombre de usuario LDAP que se ha de utilizar para conectarse al servidor LDAP.
crl.ldap.pass	*contraseña_ldap*	La contraseña LDAP asociada a crl.ldap.user.

Puede especificar varios nombres de host y puertos del servidor LDAP como se indica a continuación:

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

Puede especificar hasta 10 nombres de host. Si no especifica un número de puerto para sus servidores LDAP, se utiliza el número de puerto especificado mediante crl.ldap.port. Cada servidor LDAP debe utilizar la misma combinación de crl.ldap.user/password para el acceso.

Cuando se especifica CRLFILE DD, se carga la configuración durante la inicialización del espacio de direcciones de Advanced Message Security y se habilita la comprobación de CRL. Si no se especifica CRLFILE DD, o si el archivo de configuración CRL no está disponible o no es válida, se inhabilita la comprobación CRL.

AMS realiza una comprobación CRL utilizando los servicios de validación de certificados de IBM System SSL del modo siguiente:

Tabla 106. Comprobaciones CRL de Advanced Message Security		
Operación	**Calidad de protección**	**Certificado(s) comprobado(s)**
PUT	Privacidad	Destinatario(s)
GET	Integridad/Privacidad	Emisor

Si una operación de mensaje no pasa la comprobación CRL, Advanced Message Security realiza las acciones siguientes:

Tabla 107. Comportamiento de error de comprobación CRL de Advanced Message Security	
Operación	**Error de comprobación CRL**
PUT	El mensaje no se coloca en la cola de destino. Se devuelve un código de finalización MQCC_FAILED y un código de razón MQRC_SECURITY_ERROR a la aplicación.

Tabla 107. Comportamiento de error de comprobación CRL de Advanced Message Security (continuación)	
Operación	**Error de comprobación CRL**
GET	Se elimina el mensaje de la cola de destino y se traslada a la cola de errores de protección del sistema. Se devuelve un código de finalización MQCC_FAILED y un código de razón MQRC_SECURITY_ERROR a la aplicación.

AMS para z/OS utiliza los servicios IBM System SSL para validar los certificados, los cuales incluyen la comprobación CRL y la comprobación de confianza.

IBM MQ utiliza un valor de seguridad en el que la validación de certificados requiere que se pueda contactar con el servidor LDAP, pero no requiere que se defina una CRL.

Nota: Es responsabilidad del administrador garantizar que los servicios LDAP relevantes estén disponibles y mantener las entradas de CRL para las autoridades certificadoras relevantes.

Configuración de la protección por contraseña de AMS para los archivos de configuración

El almacenamiento de contraseñas de claves privadas y de almacén de claves como texto sin formato supone un riesgo para la seguridad, por lo que Advanced Message Security proporciona una herramienta que puede codificar esas contraseñas utilizando la clave de un usuario.

Antes de empezar

El propietario del archivo `keystore.conf` debe asegurarse de que sólo el propietario del archivo está autorizado a leer y grabar en el archivo. La protección de contraseñas descrita en este tema es sólo una medida adicional de protección. Además, debe realizar este procedimiento en un sistema seguro.

Asegúrese de que utiliza la variante **runamscred** correcta para el tipo de cliente AMS que va a leer el archivo de configuración. Si el cliente AMS es un:

- Java , debe utilizar el mandato Java **runamscred** , que se encuentra en `<IBM MQ installation root>/java/bin`
- Cliente MQI, debe utilizar el mandato **runmqascred** de MQI que se encuentra en `<IBM MQ installation root>/bin`

Procedimiento

1. Edite los archivos `keystore.conf` para incluir toda la información necesaria, incluidas las contraseñas que requieren protección.

```
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 la clave de cifrado para cifrar las contraseñas dentro de un archivo accesible para el usuario que protege el archivo `keystore.conf` .

   Esta clave debe ser la misma clave que va a utilizar el cliente de AMS más adelante:

```
ThisIsAnExampleEncryptionKey
```

3. Ejecute el mandato **runamscred** para proteger el archivo `keystore.conf` que proporciona el archivo de claves de cifrado.

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. Verifique que el archivo `keystore.conf` se ha protegido y contiene contraseñas cifradas.

Ejemplo

El ejemplo siguiente muestra el aspecto de un archivo `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
```

Información relacionada

runamscred: proteger palabras clave 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 680 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 680, "AMS" indicates " Advanced Message Security".

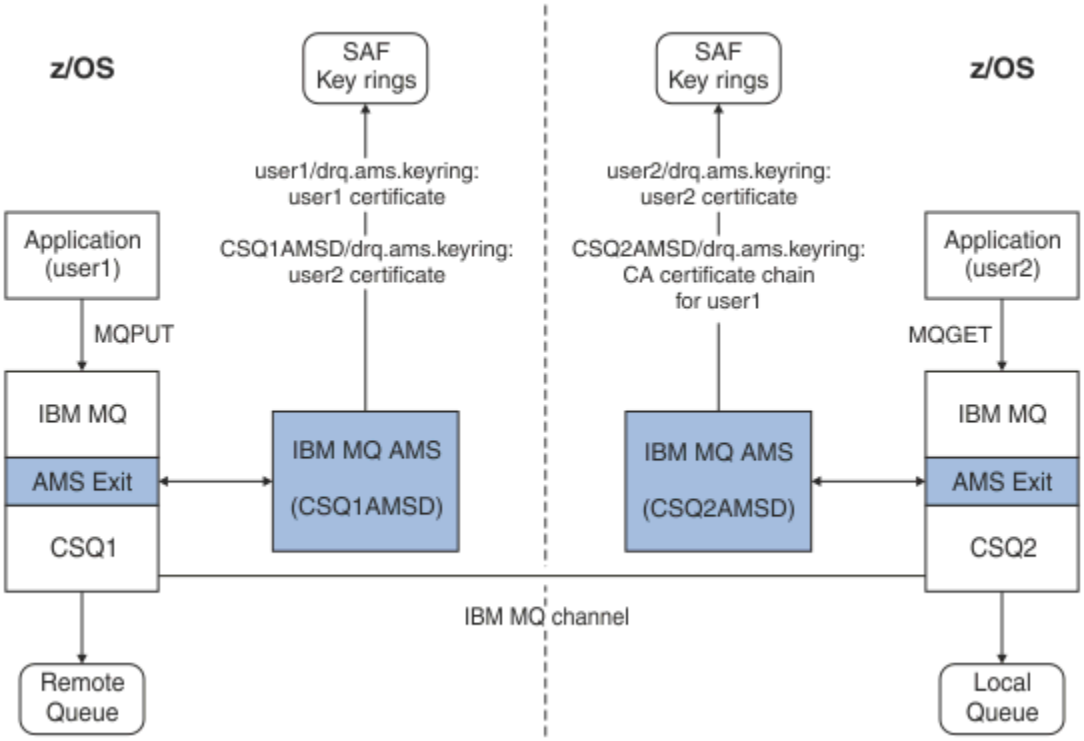


Figure 35. Application and certificate relationships

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.

Administración de políticas de seguridad de Advanced Message Security

Advanced Message Security utiliza políticas de seguridad para especificar los algoritmos criptográficos de cifrado y firma para cifrar y autenticar los mensajes que fluyen a través de las colas.

Visión general de las políticas de seguridad para AMS

Las políticas de seguridad de Advanced Message Security son objetos conceptuales que describen la forma en que un mensaje se cifra y se firma criptográficamente.

Para obtener más detalles sobre los atributos de política de seguridad, consulte los temas subordinados siguientes:

Conceptos relacionados

"Calidad de protección en AMS" en la página 686
Las políticas de protección de datos de Advanced Message Security implican una calidad de protección (QOP).

"Atributos de política de seguridad en AMS" en la página 686
Puede utilizar Advanced Message Security para seleccionar un algoritmo o método determinados para proteger los datos.

Nombres de política en AMS

El nombre de política es un nombre exclusivo que identifica una determinada política de Advanced Message Security y la cola a la que se aplica.

El nombre de política debe ser el mismo que el nombre de la cola a la que se aplica. Existe una correlación unívoca entre una política de Advanced Message Security (AMS) y una cola.

Al crear una política con el mismo nombre que el de una cola, se activa la política para dicha cola. Las colas sin nombres de política coincidentes no están protegidas por AMS.

El ámbito de la política es relevante para el gestor de colas local y sus colas. Los gestores de colas remotos deben tener sus propias políticas definidas localmente para las colas que gestionan.

Algoritmo de firma en AMS

El algoritmo de firma indica el algoritmo que se debe utilizar al firmar mensajes de datos.

Los valores válidos incluyen:

- MD5
- SHA-1
- Familia SHA-2:
  - SHA256

- SHA384 (longitud de clave mínima aceptable: 768 bits)
- SHA512 (longitud de clave mínima aceptable: 768 bits)

Una política que no especifica un algoritmo de firma, o especifica un algoritmo de NONE, implica que los mensajes colocados en la cola asociada a la política no están firmados.

Nota: La calidad de protección utilizada para colocar y recuperar mensajes debe coincidir. Si existe una discrepancia en la calidad de protección de la política entre la cola y el mensaje de la cola, el mensaje no se acepta y se envía a la cola de manejo de errores. Esta regla es válida tanto para colas locales como remotas.

Algoritmo de cifrado en AMS

El algoritmo de cifrado indica el algoritmo que debe utilizarse al cifrar los mensajes de datos colocados en la cola asociada a la política.

Los valores válidos incluyen:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

Una política que no especifica un algoritmo de cifrado o especifica un algoritmo de NONE implica que los mensajes colocados en la cola asociada a la política no están cifrados.

Tenga en cuenta que una política que especifica un algoritmo de cifrado que no sea NONE también debe especificar como mínimo un nombre distinguido de destinatario y un algoritmo de firma porque los mensajes cifrados de Advanced Message Security también están firmados.

Importante: La calidad de protección utilizada para colocar y recuperar mensajes debe coincidir. Si existe una discrepancia en la calidad de protección de la política entre la cola y el mensaje de la cola, el mensaje no se acepta y se envía a la cola de manejo de errores. Esta regla es válida tanto para colas locales como remotas.

Tolerancia en AMS

El atributo de tolerancia indica si Advanced Message Security puede aceptar mensajes sin ninguna política de seguridad especificada.

Cuando se recupera un mensaje de una cola con una política para cifrar mensajes, si el mensaje no está cifrado, se devuelve a la aplicación de llamada. Los valores válidos incluyen:

0
No (**valor predeterminado**).

1
Sí.

Una política que no especifica un valor de tolerancia o especifica 0 implica que los mensajes colocados en la cola asociada a la política deben coincidir con las reglas de política.

La tolerancia es opcional y existe para facilitar el despliegue de la configuración cuando se han aplicado políticas a las colas, pero esas colas ya contienen mensajes que no tienen una política de seguridad especificada.

Nombres distinguidos de emisor en AMS

Los nombres distinguidos de emisor identifican usuarios que están autorizados a colocar mensajes en una cola. Un remitente utiliza su certificado para firmar un mensaje, antes de colocar el mensaje en una cola.

Advanced Message Security (AMS) no comprueba si un usuario válido ha colocado un mensaje en una cola de datos protegidos hasta que se recupera el mensaje. En este momento, si la política estipula uno o más

remitentes válidos, y el usuario que ha colocado el mensaje en la cola no está en la lista de remitentes válidos, AMS devuelve un error a la aplicación receptora y coloca el mensaje en la cola de errores AMS .

Una política puede tener 0 o más nombres distinguidos de emisor válidos. Si no se especifica ningún DN de remitente para la política, cualquier remitente puede colocar mensajes protegidos por datos en la cola siempre que se confíe en el certificado del remitente. Un certificado del remitente es de confianza añadiendo el certificado público a un almacén de claves disponible para la aplicación receptora.

Los nombres distinguidos de emisor tienen el siguiente formato:

```
CN=Common Name,O=Organization,C=Country
```

Importante:

- Todos los nombres de componente DN deben estar en mayúsculas. Todos los identificadores de nombre de componente del DN deben especificarse en el orden que se muestra en la tabla siguiente:

Nombre de componente	Valor
CN	Nombre común del objeto de este nombre distinguido, tal como un nombre completo o el uso previsto de un dispositivo.
OU	Unidad dentro de la organización a la que está afiliado el objeto del nombre distinguido, tal como un departamento empresarial o un nombre de producto.
O	Organización a la que está afiliado el objeto del nombre distinguido, tal como una empresa.
L	Localidad (ciudad o municipio) donde está situado el objeto del nombre distinguido.
ST	Nombre del estado o provincia donde está situado el objeto del nombre distinguido.
C	País donde está situado el objeto del nombre distinguido.

- Si se especifica uno o más DN de emisor para la política, sólo dichos usuarios pueden colocar mensajes en la cola asociada con la política.
- Los DN de emisor, cuando se especifican, deben coincidir exactamente con el DN contenido en el certificado digital asociado con el usuario que coloca el mensaje.
- AMS permite utilizar nombres distinguidos con caracteres pertenecientes solamente al conjunto de caracteres Latin-1 Para crear DN con caracteres del conjunto, primero debe crear un certificado con un DN que se crea en la codificación UTF-8 utilizando AIX and Linux con la codificación UTF-8 activada. A continuación, debe crear una política desde una plataforma Linux o AIX con la codificación UTF-8 activada, o utilizar el plug-in AMS en IBM MQ.
- El método utilizado por AMS para convertir el nombre del remitente del formato x.509 al formato de DN siempre utiliza ST= para el valor de estado o provincia.
- Los siguientes caracteres especiales necesitan caracteres de escape:

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- Si el nombre distinguido contiene blancos intercalados, debe especificarlo entre comillas dobles.

Conceptos relacionados

"Nombres distinguidos de destinatario en AMS" en la página 685

Los nombres distinguidos de destinatario identifican a usuarios que están autorizados a recuperar mensajes de una cola.

Nombres distinguidos de destinatario en AMS

Los nombres distinguidos de destinatario identifican a usuarios que están autorizados a recuperar mensajes de una cola.

Una política puede tener 0 o más nombres distinguidos de destinatario válidos. Los nombres distinguidos de destinatario tienen el formato siguiente:

```
CN=Common Name,O=Organization,C=Country
```

Importante:

- Todos los nombres de componente DN deben estar en mayúsculas. Todos los identificadores de nombre de componente del DN deben especificarse en el orden que se muestra en la tabla siguiente:

Nombre de componente	Valor
CN	Nombre común del objeto de este nombre distinguido, tal como un nombre completo o el uso previsto de un dispositivo.
OU	Unidad dentro de la organización a la que está afiliado el objeto del nombre distinguido, tal como un departamento empresarial o un nombre de producto.
O	Organización a la que está afiliado el objeto del nombre distinguido, tal como una empresa.
L	Localidad (ciudad o municipio) donde está situado el objeto del nombre distinguido.
ST	Nombre del estado o provincia donde está situado el objeto del nombre distinguido.
C	País donde está situado el objeto del nombre distinguido.

- Si no se especifica ningún DN de destinatario para la política, cualquier usuario podrá obtener mensajes de la cola asociada con la política.
- Si se especifica uno o más DN de destinatario para la política, sólo dichos usuarios pueden obtener mensajes de la cola asociada con la política.
- Los DN de destinatario, cuando se especifican, deben coincidir exactamente con el DN contenido en el certificado digital asociado con el usuario que obtiene el mensaje.
- Advanced Message Security permite utilizar nombres distinguidos con caracteres pertenecientes solamente al conjunto de caracteres Latin-1 Para crear DN con caracteres del conjunto, primero debe crear un certificado con un DN que se crea en la codificación UTF-8 utilizando AIX o Linux con la codificación UTF-8 activada. A continuación, debe crear una política desde una plataforma Linux o AIX con la codificación UTF-8 activada o utilizar el plug-in Advanced Message Security en IBM MQ.

Conceptos relacionados

"Nombres distinguidos de emisor en AMS" en la página 683

Los nombres distinguidos de emisor identifican usuarios que están autorizados a colocar mensajes en una cola. Un remitente utiliza su certificado para firmar un mensaje, antes de colocar el mensaje en una cola.

Atributos de política de seguridad en AMS

Puede utilizar Advanced Message Security para seleccionar un algoritmo o método determinados para proteger los datos.

Una política de seguridad es un objeto conceptual que describe la forma en que un mensaje se cifra y firma criptográficamente.

Tabla 109. Atributos de política de seguridad en AMS

Atributos	**Descripción**
Nombre de política	Nombre exclusivo de la política para un gestor de colas.
Algoritmo de firma	Algoritmo criptográfico que se utiliza para firmar mensajes antes de enviarlos.
Algoritmo de cifrado	Algoritmo criptográfico que se utiliza para cifrar mensajes antes de enviarlos.
Lista de destinatarios	Lista de nombres distinguidos (DN) de certificado de posibles receptores de un mensaje.
Lista de comprobación de nombres distinguidos de firma	Lista de nombres distinguidos de firma que se deben validar durante la recuperación de mensajes.

En Advanced Message Security, los mensajes se cifran con una clave simétrica, y la clave simétrica se cifra con las claves públicas de los destinatarios. Las claves públicas se cifran con el algoritmo RSA, con claves que tienen una longitud efectiva máxima de 2048 bits. El cifrado de clave asimétrica real depende de la longitud de la clave de certificado.

Los algoritmos de clave simétrica que se pueden utilizar son los siguientes:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

Advanced Message Security también puede utilizar las funciones hash criptográficas siguientes:

- Deprecated MD5
- Deprecated SHA-1
- Familia SHA-2:
  - SHA256
  - SHA384 (longitud de clave mínima aceptable: 768 bits)
  - SHA512 (longitud de clave mínima aceptable: 768 bits)

Nota: La calidad de protección utilizada para colocar y recuperar mensajes debe coincidir. Si existe una discrepancia en la calidad de protección de la política entre la cola y el mensaje de la cola, el mensaje no se acepta y se envía a la cola de manejo de errores. Esta regla es válida tanto para colas locales como remotas.

Calidad de protección en AMS

Las políticas de protección de datos de Advanced Message Security implican una calidad de protección (QOP).

Los tres niveles de calidad de protección en Advanced Message Security se complementan con un cuarto nivel en IBM MQ 9.0 y posterior, y todo depende de los algoritmos criptográficos que se utilizan para firmar y cifrar el mensaje.

- Privacidad - los mensajes colocados en la cola deben estar firmados y cifrados.
- Integridad - los mensajes colocados en la cola deben estar firmados por el emisor.
- Confidencialidad - los mensajes colocados en la cola deben estar cifrados. Para obtener más información, consulte “Calidades de protección disponibles con AMS” en la página 610
- Ninguna - no se aplica ninguna protección de datos.

Una política que establece que los mensajes deben estar firmados cuando se colocan en una cola tiene una calidad de protección INTEGRITY. La calidad de protección INTEGRITY significa que una política estipula un algoritmo de firma, pero no estipula un algoritmo de cifrado. Los mensajes protegidos por integridad se denominan también mensajes firmados ("SIGNED").

Una política que establece que los mensajes deben estar firmados y cifrados cuando se colocan en una cola tiene una calidad de protección PRIVACY. La calidad de protección PRIVACY significa que una política estipula un algoritmo de firma y un algoritmo de cifrado. Los mensajes protegidos por privacidad se denominan también mensajes sellados ("SEALED").

Una política que establece que los mensajes deben estar cifrados cuando se colocan en una cola tiene una calidad de protección CONFIDENTIALITY. La calidad de protección CONFIDENTIALITY significa que una política estipula un algoritmo de cifrado.

Una política que no estipula un algoritmo de firma ni un algoritmo de cifrado tiene una calidad de protección NONE. Advanced Message Security no proporciona ninguna protección de datos para las colas que tienen una política cuya calidad de protección es NONE.

Gestión de políticas de seguridad en AMS

Una política de seguridad es un objeto conceptual que describe la forma en que un mensaje se cifra y firma criptográficamente.

La ubicación desde la que se ejecutan todas las tareas administrativas relacionadas con las políticas de seguridad depende de la plataforma que se utiliza.

- ALW En AIX, Linux, and Windows, utilice los mandatos DELETE POLICY, DISPLAY POLICYy SET POLICY (o PCF equivalente) para gestionar las políticas de seguridad.
  - Linux AIX En AIX and Linux, las tareas administrativas se pueden ejecutar desde *MQ_INSTALLATION_PATH*/bin.
  - Windows En las plataformas Windows, las tareas administrativas se pueden ejecutar desde cualquier ubicación, ya que las variables de entorno PATH se actualizan durante la instalación.
- IBM i En IBM i, los mandatos DSPMQMSPL, SETMQMSPL y WRKMQMSPL se instalan en la biblioteca del sistema QSYS para el idioma principal del sistema cuando se instala IBM MQ.

  Las versiones traducidas adicionales se instalan en bibliotecas QSYS29xx de acuerdo con la carga de características de idioma. Por ejemplo, una máquina que tiene inglés de Estados Unidos como idioma principal y coreano como idioma secundario tiene instalados los mandatos en inglés de Estados Unidos en QSYS y la carga de idioma secundario coreano en QSYS2962 ya que 2962 es la carga de idioma para coreano.
- z/OS En z/OS, los mandatos administrativos se ejecutan utilizando el programa de utilidad de política de seguridad de mensajes (CSQ0UTIL). Cuando se crean, modifican o suprimen políticas en z/OS, Advanced Message Security no reconoce los cambios hasta que se detiene y reinicia o se utiliza el

mandato MODIFY de z/OS para renovar la configuración de políticas de Advanced Message Security. Por ejemplo:

```
F <qmgr ssid>AMSM,REFRESH POLICY
```

Tareas relacionadas

“Creación de políticas de seguridad en AMS” en la página 688
Las políticas de seguridad definen la forma en que se protege un mensaje cuando se coloca en una cola o cuando se recibe.

“Modificación de políticas de seguridad en AMS” en la página 689
Puede utilizar Advanced Message Security para modificar los detalles de las políticas de seguridad que ya ha definido.

“Visualización y volcado de las políticas de seguridad en AMS” en la página 690
Utilice el mandato **dspmqspl** para visualizar una lista de todas las políticas de seguridad o detalles de una política con nombre de acuerdo con los parámetros que proporcione en la línea de mandatos.

“Eliminación de políticas de seguridad en AMS” en la página 691
Para eliminar las políticas de seguridad en Advanced Message Security, debe utilizar el mandato `setmqspl`.

Operando Advanced Message Security

Referencia relacionada

El programa de utilidad de política de seguridad de mensajes (CSQ0UTIL)

Creación de políticas de seguridad en AMS

Las políticas de seguridad definen la forma en que se protege un mensaje cuando se coloca en una cola o cuando se recibe.

Antes de empezar

Existen algunas condiciones básicas que se deben cumplir al crear las políticas de seguridad:

- El gestor de colas debe estar en ejecución.
- El nombre de una política de seguridad debe seguir las Reglas para denominar objetos de IBM MQ.
- Debe tener la autorización necesaria para conectarse al gestor de colas y crear una política de seguridad:
  - z/OS En z/OS, otorgue las autorizaciones documentadas en El programa de utilidad de política de seguridad de mensajes (CSQ0UTIL).
  - Multi En Multiplatforms, debe otorgar las autorizaciones `+connect`, `+inq` y `+chg` necesarias utilizando el mandato **setmqaut** .

  Para obtener más información acerca de cómo configurar la seguridad, consulte el apartado “Configuración de seguridad” en la página 137.
- z/OS En z/OS, asegúrese de que los objetos del sistema necesarios se hayan definido de acuerdo con las definiciones contenidas en CSQ4INSM.

Ejemplo

A continuación se muestra un ejemplo de creación de una política en el gestor de colas `QMGR`. La política especifica que los mensajes se firmen utilizando el algoritmo SHA256 y se cifren utilizando el algoritmo AES256 para los certificados con DN: CN=joe,O=IBM,C=US y DN: CN=jane,O=IBM,C=US. Esta política está asociada a `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
```

A continuación se muestra un ejemplo de creación de una política en el gestor de colas QMGR. La política especifica que los mensajes se cifren utilizando el algoritmo 3DES para los certificados con los nombres distinguidos: CN=john, O=IBM,C=US y CN=jeff,O=IBM,C=US y firmados con el algoritmo SHA256 para el certificado con el nombre distinguido: 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:

- La calidad de protección utilizada para colocar y recuperar mensajes debe coincidir. Si la calidad de protección de la política que se ha definido para el mensaje es más débil que la definida para una cola, el mensaje se envía a la cola de manejo de errores. Esta política es válida tanto para colas locales como remotas.

Referencia relacionada

Lista completa de los atributos del mandato setmqspl

Modificación de políticas de seguridad en AMS

Puede utilizar Advanced Message Security para modificar los detalles de las políticas de seguridad que ya ha definido.

Antes de empezar

- El gestor de colas con el que desee trabajar debe estar en ejecución.
- Debe tener la autorización necesaria para conectarse al gestor de colas y crear una política de seguridad.
  - z/OS En z/OS, otorgue las autorizaciones documentadas en El programa de utilidad de política de seguridad de mensajes (CSQ0UTIL).
  - Multi En Multiplatforms, debe otorgar las autorizaciones +connect, +inq y +chg necesarias utilizando el mandato **setmqaut** .

  Para obtener más información acerca de cómo configurar la seguridad, consulte el apartado “Configuración de seguridad” en la página 137.

Acerca de esta tarea

Para cambiar las políticas de seguridad, aplique el mandato setmqspl a una política ya existente proporcionando nuevos atributos.

Ejemplo

A continuación se muestra un ejemplo de creación de una política denominada MYQUEUE en un gestor de colas denominado QMGR, que especifica que los mensajes se van a cifrar utilizando el algoritmo 3DES para autores (**-a**) que tienen certificados con el nombre distinguido (DN) CN=alice, O=IBM, C=US y firmado con el algoritmo SHA256 para destinatarios (**-r**) que tienen certificados con el DN 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 modificar esta política, emita el mandato setmqspl con todos los atributos del ejemplo cambiando sólo los valores que desea modificar. En este ejemplo, una política creada previamente se asocia a una nueva cola y su algoritmo de cifrado se cambia a AES256:

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

Referencia relacionada

setmqspl (establecer política de seguridad)

Visualización y volcado de las políticas de seguridad en AMS

Utilice el mandato **dspmqspl** para visualizar una lista de todas las políticas de seguridad o detalles de una política con nombre de acuerdo con los parámetros que proporcione en la línea de mandatos.

Antes de empezar

- Para visualizar los detalles de las políticas de seguridad, el gestor de colas debe existir y estar en ejecución.
- Debe tener la autorización necesaria para conectarse al gestor de colas y crear una política de seguridad.
  - z/OS En z/OS, otorgue las autorizaciones documentadas en El programa de utilidad de política de seguridad de mensajes (CSQ0UTIL).
  - Multi En Multiplatforms, debe otorgar las autorizaciones `+connect`, `+inq` y `+chg` necesarias utilizando el mandato **setmqaut** .

  Para obtener más información acerca de cómo configurar la seguridad, consulte el apartado “Configuración de seguridad” en la página 137.

Acerca de esta tarea

A continuación se muestra una lista de los distintivos del mandato **dspmqspl**:

*Tabla 110. Distintivos del mandato **dspmqspl**.*

Distintivo del mandato	Explicación
-m	Nombre del gestor de colas (obligatorio).
-p	Nombre de política.
-export	La adición de este distintivo genera datos de salida que se pueden aplicar fácilmente a un gestor de colas diferente.

Ejemplo

En el ejemplo siguiente se muestra cómo crear dos políticas de seguridad 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 ejemplo muestra un mandato que muestra detalles de todas las políticas definidas para `venus.queue.manager` y el resultado que produce:

```
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 ejemplo muestra un mandato que muestra detalles de una política de seguridad seleccionada definida para `venus.queue.manager` y el resultado que produce:

```
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
```

En el ejemplo siguiente, en primer lugar debemos crear una política de seguridad y, a continuación, exportar la política utilizando el distintivo **`-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 En z/OS, la información de la política exportada la graba CSQ0UTIL en EXPORT DD.

Multi En Multiplatforms, redirija la salida a un archivo, por ejemplo:

```
dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

Para importar una política de seguridad:

- Linux AIX En AIX and Linux:
  1. Inicie la sesión como un usuario que pertenece al grupo de administración `mqm` IBM MQ.
  2. Emita `. policies.sh`.
- Windows En Windows, ejecute `policies.bat`.
- z/OS En z/OS utilice el programa de utilidad CSQ0UTIL, especificando en SYSIN el conjunto de datos que contiene la información de la política exportada.

Referencia relacionada

Lista completa de los atributos del mandato dspmqspl

Eliminación de políticas de seguridad en AMS

Para eliminar las políticas de seguridad en Advanced Message Security, debe utilizar el mandato `setmqspl`.

Antes de empezar

Existen algunas condiciones básicas que se deben cumplir al gestionar las políticas de seguridad:

- El gestor de colas debe estar en ejecución.
- Debe tener la autorización necesaria para conectarse al gestor de colas y crear una política de seguridad.
  - z/OS En z/OS, otorgue las autorizaciones documentadas en El programa de utilidad de política de seguridad de mensajes (CSQ0UTIL).

– Multi En Multiplatforms, debe otorgar las autorizaciones +connect, +inq y +chg necesarias utilizando el mandato **setmqaut** .

Para obtener más información acerca de cómo configurar la seguridad, consulte el apartado “Configuración de seguridad” en la página 137.

Acerca de esta tarea

Utilice el mandato **setmqspl** con la opción **-remove**.

Ejemplo

A continuación se muestra un ejemplo de eliminación de una política:

```
setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

Referencia relacionada

Lista completa de los atributos del mandato setmqspl

Protección de colas del sistema en AMS

Las colas del sistema permiten la comunicación entre IBM MQ y sus aplicaciones auxiliares. Cada vez que se crea un gestor de colas, se crea también una cola del sistema para almacenar mensajes y datos internos de IBM MQ. Puede proteger colas del sistema con Advanced Message Security para que solamente los usuarios autorizados puedan acceder a ellas o descifrarlas.

La protección de colas del sistema sigue el mismo patrón que la protección de colas normales. Consulte “Creación de políticas de seguridad en AMS” en la página 688.

Windows Para utilizar la protección de colas del sistema en Windows, copie el archivo keystore.conf en el directorio siguiente:

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS En z/OS, para proporcionar protección para SYSTEM.ADMIN.COMMAND.QUEUE, el servidor de mandatos debe tener acceso a keystore y keystore.conf, que contienen claves y una configuración para que el servidor de mandatos pueda acceder a claves y certificados. Todos los cambios realizados en la política de seguridad de SYSTEM.ADMIN.COMMAND.QUEUE requieren reiniciar el servidor de mandatos.

Todos los mensajes que se intercambian con la cola de mandatos se firman o se firman y cifran dependiendo de los valores de la política. Si un administrador define firmantes autorizados, el servidor de mandatos no ejecuta los mensajes de mandatos que no pasan la comprobación de nombre distinguido (DN) del firmante y no se direccionan a la cola de manejo de errores de Advanced Message Security. Los mensajes que se envían como respuestas a colas dinámicas temporales de IBM MQ Explorer no están protegidos por AMS.

Las políticas de seguridad no afectan a las colas del sistema siguientes:

- 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 Colas de transmisión y AMS

Es posible transmitir mensajes protegidos de Advanced Message Security (AMS) duplicados.

Si una cola tiene definida una política AMS que hace que los mensajes colocados en dicha cola se firmen y/o se cifren, también puede configurar el atributo **STREAMQ** de la cola para colocar una copia de cada mensaje protegido en una segunda cola. El mensaje duplicado, en modalidad continua, se firma y/o se cifra utilizando la misma política que se ha configurado para la cola original.

En el ejemplo siguiente está configurando dos colas, QUEUE1 y QUEUE2. QUEUE1 tiene su atributo **STREAMQ** configurado para transferir mensajes en modalidad continua a QUEUE2:

```
DEFINE QLOCAL(QUEUE2)

DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

Un usuario con el certificado `CN=bob,O=IBM,C=GB`pone los mensajes protegidos de AMS en QUEUE1 .

Una aplicación con el certificado `CN=alice,O=IBM,C=GB` va a consumir los mensajes de QUEUE1. Una aplicación independiente con el certificado `CN=fred,O=IBM,C=GB` va a consumir los mensajes de QUEUE2.

QUEUE1 tiene aplicada la siguiente política de privacidad de AMS :

```
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)
```

Si se ha configurado un algoritmo de cifrado en la política para QUEUE1, los destinatarios listados en la política deben incluir tanto los destinatarios de los mensajes originales de QUEUE1, como los destinatarios que van a consumir mensajes duplicados de QUEUE2.

Cuando la aplicación intenta consumir mensajes de QUEUE2 , realiza comprobaciones de integridad y/o descifra el mensaje basándose en la política que se ha establecido en QUEUE2. Si una aplicación desea consumir mensajes en modalidad continua de QUEUE2, debe establecer una política adecuada en QUEUE2 que permita comprobar la integridad de los mensajes y descifrarlos correctamente.

En concreto, el algoritmo de firma, el firmante y el algoritmo de cifrado deben ser los mismos que la política aplicada a QUEUE1. Los destinatarios de la política para QUEUE2 deben incluir la identidad del destinatario que consume el mensaje de QUEUE2.

Nota: No es necesario que la política aplicada a QUEUE2 liste todos los destinatarios nombrados en el conjunto de políticas en QUEUE1.

Por ejemplo, la política siguiente se podría establecer en QUEUE2 para permitir que una aplicación con el nombre distinguido de certificado `CN=fred,O=IBM,C=GB` lea mensajes protegidos por AMS:

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

Conceptos relacionados

Colas de modalidad continua

Otorgamiento de permisos de OAM en AMS

Los permisos de archivos autorizan a todos los usuarios ejecutar los mandatos `setmqspl` y `dspmqspl`. Sin embargo, Advanced Message Security depende del Gestor de autorizaciones sobre objetos (OAM) y todo intento de ejecutar estos mandatos por un usuario que no pertenezca al grupo mqm, que es el grupo de administración de IBM MQ, o que no tenga permisos para leer los valores de política de seguridad que se otorgan, da como resultado un error.

Procedimiento

Para otorgar los permisos necesarios a un usuario, ejecute:

```
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: Sólo es necesario establecer estas autorizaciones de OAM si tiene previsto conectar clientes, al gestor de colas, utilizando Advanced Message Security 7.0.1.

Atención: Autorización para examinar SYSTEM.PROTECTION.POLICY.QUEUE no es obligatorio en todas las situaciones. IBM MQ optimiza el rendimiento almacenando en memoria caché las políticas para que no tenga que examinar los registros para obtener detalles de política en el SYSTEM.PROTECTION.POLICY.QUEUE en todos los casos.

IBM MQ no almacena en memoria caché todas las políticas disponibles. Si hay un número alto de políticas, IBM MQ almacena en memoria caché un número limitado de políticas. Por lo tanto, si el gestor de colas tiene un número bajo de políticas definidas, no es necesario proporcionar la opción de examinar a SYSTEM.PROTECTION.POLICY.QUEUE.

Sin embargo, debe otorgar autorización para examinar esta cola, en caso de que haya un gran número de políticas definidas, o si está utilizando clientes antiguos. El sistema SYSTEM.PROTECTION.ERROR.QUEUE se utiliza para colocar mensajes de error generados por el código AMS. La autorización de colocación sobre esta cola sólo se comprueba cuando se intenta colocar un mensaje de error en la cola. La autorización de colocación sobre la cola no se comprueba cuando intenta transferir u obtener un mensaje de una cola protegida por AMS.

Otorgamiento de permisos de seguridad en AMS

Cuando utilice la seguridad de recursos de mandatos, debe configurar los permisos para permitir que Advanced Message Security funcione. En este tema se utilizan mandatos RACF en los ejemplos. Si su empresa utiliza un gestor de seguridad externo (ESM) diferente, debe utilizar los mandatos equivalentes para dicho ESM.

Existen tres aspectos para conceder permisos de seguridad:

- “El espacio de direcciones AMSM” en la página 695
- “CSQ0UTIL” en la página 696
- “Utilización de colas que tienen definida una política de Advanced Message Security” en la página 696

Notas: Los mandatos de ejemplo utilizan las variables siguientes.

1. *NombreGestColas*: El nombre del gestor de colas.

   z/OS En z/OS, este valor puede ser el nombre de un grupo de compartición de colas.
2. *nombre_usuario*: Este valor puede ser un nombre de grupo.
3. Los ejemplos muestran la clase MQQUEUE. z/OS También puede ser MXQUEUE, GMQQUEUE o GMXQUEUE. En “Profiles for queue security” en la página 208 encontrará más información.

Además, si ya existe el perfil, no necesita el mandato RDEFINE.

El espacio de direcciones AMSM

Necesita emitir alguna seguridad de IBM MQ para el nombre de usuario bajo el que se ejecuta el espacio de direcciones de Advanced Message Security.

- Para la conexión por lotes con el gestor de colas, emita:

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- Para el acceso a 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

El programa de utilidad que permite a los usuarios ejecutar mandatos **setmqspl** y **dspmqspl** requiere los siguientes permisos, donde el nombre de usuario es el ID de usuario del trabajo:

- Para la conexión por lotes con el gestor de colas, emita:

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
             PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- Para acceder a SYSTEM.PROTECTION.POLICY.QUEUE, lo cual es necesario para el mandato **setmqpol**, emita:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
              PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- Para acceder a SYSTEM.PROTECTION.POLICY.QUEUE, lo cual es necesario para el mandato **dspmqpol**, emita:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
                PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

Utilización de colas que tienen definida una política de Advanced Message Security

Cuando una aplicación no realiza ningún trabajo con colas que tienen definida una política, dicha aplicación requiere permisos adicionales para permitir que Advanced Message Security proteja los mensajes.

La aplicación requiere:

- Acceso de lectura a SYSTEM.PROTECTION.POLICY.QUEUE. Esto se lleva a cabo, emitiendo:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
          PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- Acceso de transferencia a SYSTEM.PROTECTION.ERROR.QUEUE. Esto se lleva a cabo, emitiendo:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
           PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

IBM i Configuración de certificados y el archivo de configuración del almacén de claves para AMS en IBM i

La primera tarea al configurar la protección de Advanced Message Security es crear un certificado y asociarlo con el entorno. La asociación se configura a través de un archivo retenido en el sistema de archivos integrado (IFS).

Procedimiento

1. Para crear un certificado autofirmado utilizando el conjunto de herramientas OpenSSL que se entrega con IBM i, emita el siguiente mandato desde QShell:

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
   $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
   ```

   El mandato solicita diversos atributos de nombre distinguido para un nuevo certificado autofirmado, incluidos:

   - Common Name (CN=)

- Organization (O=)
- Country (C=)

Esto crea una clave privada sin cifrar y un certificado coincidente, los dos en formato PEM (Privacy Enhanced Mail).

Para simplificar, especifique solo los valores para el nombre común, organización y país. Estos atributos y valores son importantes al crear una política.

Las solicitudes y los atributos adicionales se pueden personalizar especificando un archivo de configuración openssl personalizado en la línea de mandatos con el parámetro **-config**. Consulte la documentación de OpenSSL para obtener más detalles sobre la sintaxis del archivo de configuración.

Por ejemplo, el mandato siguiente añade extensiones adicionales de 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
```

donde myconfig.cnf es un archivo de corriente ASCII que contiene lo siguiente:

```
[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. AMS requiere que tanto el certificado como la clave privada se mantengan en el mismo archivo. Emita el siguiente mandato para hacerlo:

```
cat $HOME/mycert.pem >> $HOME/private.pem
```

El archivo `private.pem` en $HOME ahora contiene una clave privada y un certificado coincidentes, mientras que el archivo `mycert.pem` contiene todos los certificados públicos para los que puede cifrar mensajes y validar firmas.

Es necesario asociar los dos archivos con su entorno creando un archivo de configuración de almacén de claves, `keystore.conf`, en la ubicación predeterminada.

De forma predeterminada, AMS busca la configuración del almacén de claves en un subdirectorio `.mqs` del directorio de inicio.

3. En QShell, cree el archivo `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 *Creación de una política para AMS en IBM i*

Antes de crear una política, es necesario crear una cola para retener los mensajes protegidos.

Procedimiento

1. En un indicador de línea de mandatos escriba:

   ```
   CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
   ```

   donde mqmname es el nombre del gestor de colas.

   Utilice el mandato DSPMQM para comprobar que el gestor de colas es capaz de utilizar políticas de seguridad. Asegúrese de que **Security Policy Capability** muestra *YES*.

   La política más simple que puede definir es una política de integridad, que se consigue creando una política con un algoritmo de firma digital, pero sin algoritmo de cifrado.

   Los mensajes están firmados pero no cifrados. Si los mensajes se van a cifrar, debe especificar un algoritmo de cifrado y uno o más destinatarios de mensajes que desee.

   Un certificado del almacén de claves público para un destinatario de mensaje que desee se identifica mediante un nombre distinguido.
2. Visualice los nombres distinguidos de los certificados en el almacén de claves público, mycert.pem en $HOME, utilizando el siguiente mandato en QShell:

   ```
   /QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
   ```

   Es necesario especificar el nombre distinguido como un destinatario deseado y el nombre de política debe coincidir con el nombre de cola que debe protegerse.
3. Por ejemplo, en un indicador de mandatos CL escriba:

   ```
   SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
   O=.., C=..')
   ```

   donde mqmname es el nombre del gestor de colas.

   Una vez que se haya creado la política, todos los mensajes que se colocan, examinan o eliminan de modo destructivo a través de dicho nombre de cola están sujetos a la política de AMS.

Referencia relacionada

Visualizar gestor de colas de mensajes (DSPMQM)

Establecer política de seguridad de MQM (SETMQMSPL)

IBM i *Prueba de una política para AMS en IBM i*

Utilice las aplicaciones de ejemplo proporcionadas con el producto para probar las políticas de seguridad.

Acerca de esta tarea

Puede utilizar las aplicaciones de ejemplo proporcionadas con IBM MQ, como AMQSPUT4, AMQSGET4, AMQSGBR4 y las herramientas como WRKMQMMSG para transferir, examinar y obtener mensajes utilizando el nombre de cola PROTECTED.

Siempre y cuando todo se haya configurado correctamente, no debería haber ninguna diferencia en el comportamiento de la aplicación con el de una cola no protegida para este usuario.

Sin embargo, un usuario no configurado para Advanced Message Security o un usuario que no tenga la clave privada necesaria para descifrar el mensaje, no podrá ver el mensaje. El usuario recibe un código de terminación de RCFAIL, equivalente a MQCC_FAILED (2) y un código de razón de RC2063 (MQRC_SECURITY_ERROR).

Para ver que la protección AMS está en vigor, transfiera algunos mensajes de prueba a la cola PROTECTED, por ejemplo utilizando AMQSPUT0. Entonces podrá crear una cola de alias para examinar los datos protegidos sin formato mientras se está en reposo.

Procedimiento

Para otorgar los permisos necesarios a un usuario, ejecute:

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

Examinar utilizando el nombre de cola ALIAS, por ejemplo utilizando AMQSBCG4 o WRKMQMMSG, debería revelar mensajes `scrambled` más grandes donde un examen de la cola PROTECTED muestra mensajes de texto simple.

Los mensajes mezclados son visibles pero el texto simple original no puede descifrarse utilizando la cola ALIAS, ya que no hay política para que AMS fuerce la coincidencia de este nombre. Por lo tanto, se devuelven los datos protegidos sin formato.

Referencia relacionada

Establecer política de seguridad de MQM (SETMQMSPL)

Trabajar con mensajes MQ (WRKMQMMSG)

Sucesos de mandato y configuración para AMS

Con Advanced Message Security, puede generar mensajes para sucesos de mandato y de configuración, que se pueden registrar y servir como registro de los cambios de política con fines de auditoría.

Los sucesos de mandatos y configuración que genera IBM MQ son mensajes en formato PCF enviados a las colas dedicadas del gestor de colas donde se produce el suceso.

Los mensajes de sucesos de configuración se envían a la cola SYSTEM.ADMIN.CONFIG.EVENT.

Los mensajes de sucesos de mandatos se envían a la cola SYSTEM.ADMIN.COMMAND.EVENT.

Los sucesos se generan con independencia de las herramientas que utilice para gestionar las políticas de seguridad de Advanced Message Security.

En Advanced Message Security, existen cuatro tipos de sucesos generados por distintas acciones en políticas de seguridad:

- "Creación de políticas de seguridad en AMS" en la página 688, que produce dos mensajes de suceso de IBM MQ:
  - Un suceso de configuración
  - Un suceso de mandato
- "Modificación de políticas de seguridad en AMS" en la página 689, que produce tres mensajes de suceso de IBM MQ:
  - Un suceso de configuración que contiene valores antiguos de política de seguridad
  - Un suceso de configuración que contiene valores nuevos de política de seguridad
  - Un suceso de mandato
- "Visualización y volcado de las políticas de seguridad en AMS" en la página 690, que produce un solo mensajes de suceso de IBM MQ:
  - Un suceso de mandato
- "Eliminación de políticas de seguridad en AMS" en la página 691, que produce dos mensajes de suceso de IBM MQ:
  - Un suceso de configuración
  - Un suceso de mandato

Habilitación e inhabilitación del registro de sucesos para AMS

Puede controlar sucesos de mandato y de configuración mediante los atributos del gestor de colas **CONFIGEV** y **CMDEV**. Para habilitar estos sucesos, establezca el atributo de gestor de colas adecuado

en ENABLED. Para inhabilitar estos sucesos, establezca el atributo adecuado del gestor de colas en DISABLED.

Procedimiento

Sucesos de configuración

Para habilitar los sucesos de configuración, establezca **CONFIGEV** en ENABLED. Para inhabilitar los sucesos de configuración, establezca **CONFIGEV** en DISABLED. Por ejemplo, puede habilitar los sucesos de configuración mediante el mandato MQSC siguiente:

```
ALTER QMGR CONFIGEV (ENABLED)
```

Sucesos de mandatos

Para habilitar los sucesos de mandatos, establezca **CMDEV** en ENABLED. Para habilitar los sucesos de mandato para todos los mandatos, excepto los mandatos **DISPLAY MQSC** y los mandatos Inquire PCF, establezca **CMDEV** en NODISPLAY. Para inhabilitar los sucesos de mandato, establezca **CMDEV** en DISABLED. Por ejemplo, puede habilitar los sucesos de mandato mediante el mandato MQSC siguiente:

```
ALTER QMGR CMDEV (ENABLED)
```

Tareas relacionadas

Control de sucesos de configuración, mandato y registro en IBM MQ

Formato de mensaje de suceso de mandato para AMS

El mensaje de suceso de mandato consta de la estructura MQCFH y los parámetros PCF que le siguen a continuación.

Estos son valores de MQCFH seleccionados:

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

Nota: El valor de ParameterCount es dos porque siempre hay dos parámetros de tipo MQCFGR (grupo). Cada grupo consta de parámetros adecuados. Los datos de suceso constan de dos grupos, CommandContext y CommandData.

CommandContext contiene:

EventUserID

Descripción:	El ID de usuario que ha emitido el mandato o la llamada que ha generado el suceso. (Éste es el mismo ID de usuario que se utiliza para comprobar la autorización para poder emitir el mandato o la llamada; para los mandatos recibidos de una cola, éste es también el identificador de usuario (UserIdentifier) del MD del mensaje de mandato).
Identificador:	MQCACF_EVENT_USER_ID.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_USER_ID_LENGTH.
Se devuelve:	Siempre.

EventOrigin

Descripción: El origen de la acción que ha provocado el suceso.

Identificador: MQIACF_EVENT_ORIGIN.

Tipo de datos: MQCFIN.

Valores:

MQEVO_CONSOLE
Mandato de consola - línea de mandatos.

MQEVO_MSG
Mensaje de mandato del plugin IBM MQ Explorer.

Se devuelve: Siempre.

EventQMgr

Descripción: El gestor de colas en el que se introdujo el mandato o la llamada. (El gestor de colas donde se ejecuta el mandato y que genera el suceso se encuentra en el MD del mensaje de suceso).

Identificador: MQCACF_EVENT_Q_MGR.

Tipo de datos: MQCFST.

Longitud máxima: MQ_Q_MGR_NAME_LENGTH.

Se devuelve: Siempre.

EventAccountingToken

Descripción: Para los mandatos recibidos como mensaje (MQEVO_MSG), es el token contable (AccountingToken) del MD del mensaje de mandato.

Identificador: MQBACF_EVENT_ACCOUNTING_TOKEN.

Tipo de datos: MQCFBS.

Longitud máxima: MQ_ACCOUNTING_TOKEN_LENGTH.

Se devuelve: Sólo si EventOrigin es MQEVO_MSG.

EventIdentityData

Descripción: Para los mandatos recibidos como mensaje (MQEVO_MSG), son los datos de identidad de la aplicación (ApplIdentityData) del MD del mensaje de mandato.

Identificador: MQCACF_EVENT_APPL_IDENTITY.

Tipo de datos: MQCFST.

Longitud máxima: MQ_APPL_IDENTITY_DATA_LENGTH.

Se devuelve: Sólo si EventOrigin es MQEVO_MSG.

EventApplType

Descripción: Para los mandatos recibidos como mensaje (MQEVO_MSG), es el tipo de aplicación (PutApplType) del MD del mensaje de mandato.

Identificador: MQIACF_EVENT_APPL_TYPE.

Tipo de datos: MQCFIN.

Se devuelve: Sólo si EventOrigin es MQEVO_MSG.

EventApplName

Descripción:	Para los mandatos recibidos como mensaje (MQEVO_MSG), es el nombre de la aplicación (PutApplName) del MD del mensaje de mandato.
Identificador:	MQCACF_EVENT_APPL_NAME.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_APPL_NAME_LENGTH.
Se devuelve:	Sólo si EventOrigin es MQEVO_MSG.

EventApplOrigin

Descripción:	Para los mandatos recibidos como mensaje (MQEVO_MSG), son los datos de origen de la aplicación (ApplOriginData) del MD del mensaje de mandato.
Identificador:	MQCACF_EVENT_APPL_ORIGIN.
Tipo de datos:	MQCFST.
Longitud máxima:	MQ_APPL_ORIGIN_DATA_LENGTH.
Se devuelve:	Sólo si EventOrigin es MQEVO_MSG.

Mandato

Descripción:	El código del mandato.
Identificador:	MQIACF_COMMAND.
Tipo de datos:	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** Estos se definen en IBM MQ 8.0 `cmqcfc.h`
Se devuelve:	Siempre.

CommandData contiene elementos PCF que conforman el mandato PCF.

Formato de mensaje de suceso de configuración para AMS

Los sucesos de configuración son mensajes PCF de formato Advanced Message Security estándar.

Los valores posibles para el descriptor de mensajes MQMD se pueden encontrar en la sección Mensaje de suceso MQMD (descriptor de mensaje).

Estos son valores de MQMD seleccionados:

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR           //for both CLI and command server
```

El almacenamiento intermedio de mensaje consta de la estructura MQCFH y la estructura de parámetro que le sigue. Los valores posibles de MQCFH se pueden encontrar en Mensaje de suceso MQCFH (cabecera PCF).

Estos son valores de MQCFH seleccionados:

```
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}
```

Los parámetros que siguen a MQCFH son:

EventUserID

Descripción: El ID de usuario que ha emitido el mandato o la llamada que ha generado el suceso. (Éste es el mismo ID de usuario que se utiliza para comprobar la autorización para poder emitir el mandato o la llamada; para los mandatos recibidos de una cola, éste es también el identificador de usuario (UserIdentifier) del MD del mensaje de mandato).

Identificador: **MQCACF_EVENT_USER_ID**

Tipo de datos: MQCFST.

Longitud máxima: MQ_USER_ID_LENGTH.

Se devuelve: Siempre.

SecurityId

Descripción: Es el valor de MQMD.AccountingToken para el mensaje del servidor de mandatos o Windows SID para el mandato local.

Identificador: **MQBACF_EVENT_SECURITY_ID**

Tipo de datos: MQCBS.

Longitud máxima: MQ_SECURITY_ID_LENGTH.

Se devuelve: Siempre.

EventOrigin

Descripción: El origen de la acción que ha provocado el suceso.

Identificador: **MQIACF_EVENT_ORIGIN**

Tipo de datos: MQCFIN.

Valores:

MQEVO_CONSOLE
Mandato de consola - línea de mandatos.

MQEVO_MSG
Mensaje de mandato del plugin IBM MQ Explorer.

Se devuelve: Siempre.

EventQMgr

Descripción: El gestor de colas en el que se introdujo el mandato o la llamada. (El gestor de colas donde se ejecuta el mandato y que genera el suceso se encuentra en el MD del mensaje de suceso).

Identificador: **MQCACF_EVENT_Q_MGR**

Tipo de datos: MQCFST

Longitud máxima: MQ_Q_MGR_NAME_LENGTH

Se devuelve: Siempre.

ObjectType

Descripción: Tipo de objeto.

Identificador: **MQIACF_OBJECT_TYPE**

Tipo de datos: MQCFIN

Valor: **MQOT_PROT_POLICY**
Política de protección de Advanced Message Security. **1019** - valor numérico definido en IBM MQ 8.0 o en el archivo `cmqc.h`.

Se devuelve: Siempre.

PolicyName

Descripción: El nombre de política de Advanced Message Security.

Identificador: **MQCA_POLICY_NAME**.

Tipo de datos: MQCFST.

Valor: **2112** - valor numérico definido en IBM MQ 8.0 o en el archivo `cmqc.h`.

Longitud máxima: MQ_OBJECT_NAME_LENGTH.

Se devuelve: Siempre.

PolicyVersion

Descripción: Versión de la política de Advanced Message Security.

Identificador: **MQIA_POLICY_VERSION**

Tipo de datos: MQCFIN

Valor **238** - valor numérico definido en IBM MQ 8.0 o en el archivo `cmqc.h`.

Se devuelve: Siempre

TolerateFlag

Descripción: Distintivo de tolerancia de política de Advanced Message Security.

Identificador: **MQIA_TOLERATE_UNPROTECTED**

Tipo de datos: MQCFIN

Valor **235** - valor numérico definido en IBM MQ 8.0 o en el archivo `cmqc.h`.

Se devuelve: Siempre.

SignatureAlgorithm

Descripción: Algoritmo de firma de política de Advanced Message Security.

Identificador: **MQIA_SIGNATURE_ALGORITHM**

Tipo de datos: MQCFIN

Valor: **236** - valor numérico definido en IBM MQ 8.0 o en el archivo `cmqc.h`.

Se devuelve: Siempre que hay un algoritmo de firma definido en la política de Advanced Message Security

EncryptionAlgorithm

Descripción: Algoritmo de cifrado de la política de Advanced Message Security.

Identificador: **MQIA_ENCRYPTION_ALGORITHM**

Tipo de datos: MQCFIN

Valor: **237** - valor numérico definido en IBM MQ 8.0 o en el archivo `cmqc.h`.

Se devuelve: Siempre que hay un algoritmo de cifrado definido en la política de IBM MQ

SignerDNs

Descripción: Nombre distinguido de los firmantes permitidos.

Identificador: **MQCA_SIGNER_DN**

Tipo de datos: MQCFSL

Valor: **2113** - valor numérico definido en IBM MQ 8.0 o en el archivo `cmqc.h`.

Longitud máxima: Nombre distinguido de firmante más largo de la política, pero no más largo que MQ_DISTINGUISHED_NAME_LENGTH

Se devuelve: Siempre que está definido en la política de IBM MQ.

RecipientDNs

Descripción: Nombre distinguido de los firmantes permitidos.

Identificador: **MQCA_RECIPIENT_DN**

Tipo de datos: MQCFSL

Valor: **2114** - valor numérico definido en IBM MQ 8.0 o en el archivo `cmqc.h`.

Longitud máxima: Nombre distinguido de destinatario más largo de la política, pero no más largo que MQ_DISTINGUISHED_NAME_LENGTH.

Se devuelve: Siempre que está definido en la política de IBM MQ.

Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en los Estados Unidos.

Es posible que IBM no ofrezca los productos, servicios o las características que se tratan en este documento en otros países. Consulte al representante local de IBM para obtener información sobre los productos y servicios que actualmente pueden adquirirse en su zona. Las referencias a programas, productos o servicios de IBM no pretenden establecer ni implicar que sólo puedan utilizarse dichos productos, programas o servicios de IBM. En su lugar podrá utilizarse cualquier producto, programa o servicio equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio no IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran el tema principal descrito en este documento. El suministro de este documento no le otorga ninguna licencia sobre estas patentes. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el Departamento de propiedad intelectual de IBM de su país o envíe las consultas por escrito a:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokio 103-8510, Japón

El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde estas disposiciones contradigan la legislación vigente: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN NINGÚN TIPO DE GARANTÍA, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INCUMPLIMIENTO, COMERCIABILIDAD O IDONEIDAD PARA UNA FINALIDAD DETERMINADA. Algunas legislaciones no contemplan la exclusión de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que puede haber usuarios a los que no les afecte dicha norma.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información aquí contenida está sometida a cambios periódicos; tales cambios se irán incorporando en nuevas ediciones de la publicación. IBM puede realizar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento sin previo aviso.

Las referencias hechas en esta publicación a sitios web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen de modo alguno un aval de esos sitios web. Los materiales de estos sitios web no forman parte de los materiales para este producto IBM, por lo que la utilización de dichos sitios web es a cuenta y riesgo del usuario.

IBM puede utilizar o distribuir cualquier información que el usuario le proporcione del modo que considere apropiado sin incurrir por ello en ninguna obligación con respeto al usuario.

Los titulares de licencias de este programa que deseen información del mismo con el fin de permitir: (i) el intercambio de información entre los programas creados de forma independiente y otros programas (incluido este) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

© Copyright IBM Corp. 2007, 2024

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluyendo, en algunos casos, el pago de una cantidad.

El programa bajo licencia que se describe en esta información y todo el material bajo licencia disponible para el mismo lo proporciona IBM bajo los términos del Acuerdo de cliente de IBM, el Acuerdo de licencia de programas internacional de IBM o cualquier acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos en este documento se han obtenido en un entorno controlado. Por consiguiente, los resultados obtenidos en otros entornos operativos pueden variar de manera significativa. Es posible que algunas mediciones se hayan realizado en sistemas en nivel de desarrollo y no existe ninguna garantía de que estas mediciones serán las mismas en sistemas disponibles generalmente. Además, es posible que algunas mediciones se hayan estimado mediante extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relativa a productos que no son de IBM se obtuvo de los proveedores de esos productos, sus anuncios publicados u otras fuentes de disponibilidad pública. IBM no ha comprobado estos productos y no puede confirmar la precisión de su rendimiento, compatibilidad o alguna reclamación relacionada con productos que no sean de IBM. Todas las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones relacionadas con una futura intención o tendencia de IBM están sujetas a cambios o se pueden retirar sin previo aviso y sólo representan metas y objetivos.

Este documento contiene ejemplos de datos e informes que se utilizan diariamente en la actividad de la empresa. Para ilustrar los ejemplos de la forma más completa posible, éstos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con los nombres y direcciones utilizados por una empresa real es puramente casual.

LICENCIA DE DERECHOS DE AUTOR:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma sin pagar ninguna cuota a IBM para fines de desarrollo, uso, marketing o distribución de programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas de ejemplo. Los ejemplos no se han probado minuciosamente bajo todas las condiciones. IBM, por tanto, no puede garantizar la fiabilidad, servicio o funciones de estos programas.

Puede que si visualiza esta información en copia software, las fotografías e ilustraciones a color no aparezcan.

Información acerca de las interfaces de programación

La información de interfaz de programación, si se proporciona, está pensada para ayudarle a crear software de aplicación para su uso con este programa.

Este manual contiene información sobre las interfaces de programación previstas que permiten al cliente escribir programas para obtener los servicios de IBM MQ.

Sin embargo, esta información puede contener también información de diagnóstico, modificación y ajustes. La información de diagnóstico, modificación y ajustes se proporciona para ayudarle a depurar el software de aplicación.

Importante: No utilice esta información de diagnóstico, modificación y ajuste como interfaz de programación porque está sujeta a cambios.

Marcas registradas

IBM, el logotipo de IBM , ibm.com, son marcas registradas de IBM Corporation, registradas en muchas jurisdicciones de todo el mundo. Hay disponible una lista actual de marcas registradas de IBM en la web en "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras empresas.

Microsoft y Windows son marcas registradas de Microsoft Corporation en Estados Unidos y/o otros países.

UNIX es una marca registrada de Open Group en Estados Unidos y en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos y en otros países.

Este producto incluye software desarrollado por Eclipse Project (https://www.eclipse.org/).

Java y todas las marcas registradas y logotipos son marcas registradas de Oracle o sus afiliados.

Número Pieza:

(1P) P/N: